



# MaCARI: une méthode d'accès déterministe et économe en énergie pour les réseaux de capteurs sans fil

Gérard Chalhoub

## ► To cite this version:

Gérard Chalhoub. MaCARI: une méthode d'accès déterministe et économe en énergie pour les réseaux de capteurs sans fil. Réseaux et télécommunications [cs.NI]. Université Blaise Pascal - Clermont-Ferrand II, 2009. Français. NNT : 2009CLF21988 . tel-00724590

**HAL Id: tel-00724590**

**<https://theses.hal.science/tel-00724590>**

Submitted on 21 Aug 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro d'ordre : D. U : 1988

E D S P I C : 466

UNIVERSITÉ BLAISE PASCAL

ÉCOLE DOCTORALE SCIENCES  
POUR L'INGÉNIEUR DE CLERMONT-FERRAND

Laboratoire d'Informatique, de Modélisation et d'Optimisation des  
Systèmes

THÈSE

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

Discipline : Informatique

# **MaCARI : Une méthode d'accès déterministe et économe en énergie pour les réseaux de capteurs sans fil**

Présentée par Gérard CHALHOUB

Directeur de thèse :	M. Michel MISSON	Professeur à l'Université de Clermont
Rapporteurs :	M. Gilles DELISLE	Professeur, Directeur du Centre d'intégration technologique de Québec
	M. Francis LEPAGE	Professeur à l'Université de Nancy
	M. David SIMPLOT-RYL	Professeur à l'Université de Lille
Examineurs :	M. Tuan DANG	Docteur, Expert Ingénieur de Re- cherche à EDF
	M. Alexandre GUITTON	Maître de Conférences à l'Université de Clermont
	Mme Pascale MINET	Chercheur HDR à l'INRIA
	M. Thierry VAL	Professeur à l'Université de Toulouse



*À ma famille...*



# Remerciements

En préambule à ce rapport de thèse, je souhaiterais adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce travail.

Je voudrais tout d'abord exprimer ma profonde gratitude à Monsieur Michel Misson, mon directeur de thèse, qui s'est toujours montré à l'écoute et très disponible tout au long de cette thèse, pour son soutien et ses encouragements continuels ainsi que sa patience. Ce que j'ai appris en travaillant avec Monsieur Misson ne se limite pas à l'aspect scientifique mais aussi humain et relationnel.

Je tiens également à remercier le LATTIS, en particulier Messieurs Thierry Val, Adrien Van Den Bossche et Erwan Livolant, avec qui nous avons fortement collaborer pour concrétiser nos contributions dans le cadre du projet OCARI.

Je souhaite remercier Messieurs Gilles Delisle, Francis Lepage et David Simplot-Ryl de m'avoir fait l'honneur d'évaluer mon travail et d'en être rapporteurs, l'échange que j'ai eu avec eux m'a beaucoup enrichi.

Mes remerciements s'adressent également à l'ensemble des partenaires du projet OCARI, avec qui j'ai eu beaucoup de plaisir à travailler, en particulier Madame Pascale Minet, chercheur HDR à l'INRIA, avec qui nous avons beaucoup collaboré et Monsieur Tuan Dang, ingénieur expert à EDF et chef du projet OCARI pour son encouragement et son enthousiasme envers notre contribution dans le projet.

Je remercie également les membres de l'équipe Réseaux et Protocoles au sein de laquelle j'ai effectué mon travail de doctorat. Je remercie en particulier Monsieur Alexandre Guitton, maître de conférences à l'université Blaise Pascal, acteur essentiel dans le projet OCARI, pour son support et son encouragement. Je remercie également Monsieur François Delobel, maître de conférence à l'université d'Auvergne, pour son travail sur l'animateur graphique, un outil qui m'a beaucoup servi.

Je tiens à exprimer ma reconnaissance envers Madame Marie-Françoise Servajean et Madame Frédérique Jacquet, maîtres de conférences à l'université d'Auvergne, qui ont eu la gentillesse de lire et corriger ce document.

Je remercie Monsieur Joël Toussaint, chef du département Réseaux et Télécommunications de l'IUT dans lequel j'effectue mon enseignement, pour son support et son aide tout au long de ma thèse.

Je n'oublie pas mes collègues du bureau, en particulier Monsieur Chakib Baouche pour sa présence et son aide pour les préparations le jour de la soutenance. Je tiens aussi à remercier Mademoiselle Cécile Bechtold pour les photos qui m'ont été très utiles pour la préparation des diapositives de la soutenance.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce travail.

# Table des matières

<b>Introduction</b>	<b>21</b>
<b>1 Les réseaux de capteurs sans fil</b>	<b>25</b>
1 Domaines d'application des réseaux de capteurs sans fil . . . . .	27
1.1 Applications militaires . . . . .	27
1.2 Applications liées à la sécurité . . . . .	27
1.3 Applications environnementales . . . . .	27
1.4 Applications médicales . . . . .	28
1.5 Applications écologiques . . . . .	28
1.6 Applications de traçabilité et de localisation . . . . .	28
1.7 Applications industrielles . . . . .	29
2 Besoins des applications industrielles . . . . .	29
2.1 Économie d'énergie . . . . .	29
2.2 Qualité de service . . . . .	30
2.3 Auto-configuration . . . . .	30
2.4 Mobilité . . . . .	30
2.5 Sécurité . . . . .	30
3 Projet OCARI . . . . .	31
3.1 Les partenaires du projet OCARI . . . . .	31
3.2 Objectifs du projet OCARI . . . . .	32
<b>2 État de l'art des protocoles MAC pour les réseaux de capteurs sans fil</b>	<b>35</b>
1 Les méthodes d'accès dans les réseaux de capteurs sans fil . . . . .	35
1.1 Contraintes et spécificités des réseaux de capteurs sans fil . . . . .	35
1.1.1 Économie d'énergie . . . . .	37
<i>Overhearing</i> . . . . .	37



	Collisions . . . . .	38
	<i>Idle listening</i> . . . . .	38
	Envois infructueux . . . . .	38
	<i>Overhead</i> . . . . .	38
1.1.2	Déterminisme . . . . .	38
1.1.3	Auto-configuration . . . . .	39
1.2	Exemples de protocoles MAC pour les réseaux de capteurs sans fil . . . . .	39
1.2.1	Protocoles basés sur un séquençement temporel . . . . .	39
	TRAMA . . . . .	39
	FLAMA . . . . .	40
	E-MAC, L-MAC et AI-LMAC . . . . .	41
1.2.2	Protocoles basés sur la contention . . . . .	42
	CSMA/CA de la norme IEEE 802.11 . . . . .	42
	<i>Clear Channel Assessment</i> . . . . .	43
	Accès au médium . . . . .	44
	Terminal caché et terminal exposé . . . . .	44
	S-MAC, T-MAC et D-MAC . . . . .	45
	LPL : B-MAC, WiseMAC, B-MAC+, X-MAC et DW-LPL . . . . .	47
1.2.3	Protocoles hybrides . . . . .	48
	Z-MAC . . . . .	48
	G-MAC . . . . .	50
	Funneling-MAC . . . . .	50
1.2.4	Récapitulatif des caractéristiques des protocoles présentés . . . . .	52
2	La norme IEEE 802.15.4/ZigBee . . . . .	53
2.1	Couche physique IEEE 802.15.4 . . . . .	55
2.1.1	Activation et désactivation du module radio . . . . .	56
2.1.2	Indication de la qualité du lien . . . . .	56
2.1.3	Test d'occupation du médium ou CCA ( <i>Clear Channel Assessment</i> ) . . . . .	56
2.1.4	Sélection du canal . . . . .	56
2.2	Couche MAC IEEE 802.15.4 . . . . .	56
2.2.1	Accès au médium . . . . .	57
	Algorithme de CSMA/CA slotté . . . . .	58
	Algorithme de CSMA/CA non-slotté . . . . .	61
2.2.2	Les <i>scans</i> , la création du réseau, les associations et la synchronisation . . . . .	63

	Les <i>scans</i> . . . . .	63
	Création du réseau . . . . .	63
	Association et désassociation . . . . .	63
	Synchronisation avec le <i>beacon</i> du coordinateur . . . . .	64
2.2.3	Échange de données . . . . .	64
	Échange direct . . . . .	64
	Échange indirect . . . . .	65
	Échange en GTS . . . . .	65
2.3	Couche réseau ZigBee . . . . .	65
2.3.1	Création de la topologie . . . . .	66
2.3.2	Allocation des adresses . . . . .	66
	Allocation d'adresses hiérarchiques . . . . .	66
	Allocation d'adresses aléatoires . . . . .	67
2.3.3	Routage . . . . .	68
	Routage hiérarchique . . . . .	68
2.4	Couche application ZigBee . . . . .	69
2.5	Problèmes persistants . . . . .	70
2.5.1	Collisions directes de <i>beacons</i> . . . . .	71
	Approche <i>Beacon Only Period</i> . . . . .	71
	Approche par supertrames asynchrones . . . . .	72
2.5.2	Collisions indirectes de <i>beacons</i> . . . . .	73
	Approche réactive . . . . .	73
	Approche proactive . . . . .	74
2.6	Synthèse . . . . .	74
<b>3</b>	<b>Notre contribution</b>	<b>75</b>
1	MaCARI . . . . .	75
1.1	Description générale de MaCARI . . . . .	76
1.1.1	Synchronisation et découpage temporel . . . . .	76
1.1.2	Encapsulation des trames . . . . .	79
1.1.3	Estampillage temporel des trames . . . . .	79
1.1.4	Accès au médium . . . . .	79
1.1.5	Création du réseau et gestion de la topologie . . . . .	80
1.1.6	Affectation des adresses hiérarchiques . . . . .	80

1.1.7	Relais garanti pour une différenciation de service . . . . .	80
1.2	La pile OCARI . . . . .	80
1.2.1	EOLSR . . . . .	81
	Interaction MaCARI-EOLSR . . . . .	82
1.2.2	SERENA . . . . .	82
	Interaction MaCARI-SERENA . . . . .	83
1.2.3	Les échanges de primitives de données entre la couche supérieure et Ma- CARI . . . . .	83
2	La période de synchronisation . . . . .	85
2.1	Description . . . . .	85
2.2	Format d'un <i>beacon</i> MaCARI . . . . .	86
2.2.1	L'en-tête du <i>beacon</i> . . . . .	86
2.2.2	Le champ <i>payload</i> du <i>beacon</i> . . . . .	87
2.2.3	Unicité de la <i>payload</i> du <i>beacon</i> . . . . .	88
2.3	Périodicité de la synchronisation . . . . .	88
2.4	Calcul de $T_1$ et de l'instant d'envoi du <i>beacon</i> . . . . .	88
2.5	L'estampillage des trames . . . . .	89
2.6	Économie d'énergie . . . . .	90
2.7	Réduction de la durée de la période de synchronisation . . . . .	91
2.7.1	Amélioration par envoi anticipé . . . . .	91
	Modification du format du <i>beacon</i> pour prendre en compte l'amélioration . . . . .	92
2.7.2	Amélioration de l'ordre de synchronisation . . . . .	93
	Parcours en largeur . . . . .	95
	Permutation . . . . .	95
3	Segmentation . . . . .	96
3.1	Description de la segmentation . . . . .	96
3.1.1	Séquencement des activités des étoiles . . . . .	96
3.1.2	Période de routage . . . . .	98
3.1.3	Relais garanti . . . . .	98
3.2	Économie d'énergie . . . . .	98
3.3	Calcul des instants des périodes d'activité . . . . .	99
3.4	Délai garanti . . . . .	99
3.4.1	Adaptation de la taille des intervalles de relais . . . . .	102
	Calcul en local selon le trafic cumulé . . . . .	102

	Calcul par le CPAN en fonction des indications de trafic et du nombre de descendants . . . . .	102
3.5	Schéma récapitulatif . . . . .	103
<b>4</b>	<b>Résultats</b>	<b>105</b>
1	Validation . . . . .	105
1.1	Simulation . . . . .	105
1.1.1	Simulation du médium sans fil . . . . .	106
	Le modèle <i>Free space</i> . . . . .	106
	Le modèle <i>Two-ray ground</i> . . . . .	107
	Le modèle <i>Shadowing</i> . . . . .	107
	Le modèle ITU-R P1238-4 . . . . .	107
	Choix et vérification du modèle de propagation . . . . .	107
1.1.2	Simulation de la couche physique . . . . .	108
	Effet de capture et collision . . . . .	108
	L'implémentation du CCA . . . . .	110
1.1.3	Simulation de la couche MAC . . . . .	111
	L'implémentation du CSMA/CA slotté . . . . .	111
1.1.4	Simulation de la couche réseau . . . . .	113
	Protocole de routage $[T_2; T_3]$ . . . . .	113
1.1.5	Aspect graphique . . . . .	113
1.2	Maquettage . . . . .	116
1.2.1	Implémentation sur des cartes B2400ZB . . . . .	116
1.2.2	Synchronisation et compteurs . . . . .	116
1.2.3	Machine d'état MaCARI et gestion des files d'attente . . . . .	118
1.2.4	Gestion des reports . . . . .	118
2	Synchronisation . . . . .	119
2.1	Simulation . . . . .	119
2.1.1	<i>Scenarii</i> de tests . . . . .	119
2.2	Mesures de précision . . . . .	120
2.2.1	<i>Scenarii</i> de tests . . . . .	121
3	Segmentation . . . . .	123
3.1	Simulation . . . . .	123
3.1.1	Évaluation de la durée des périodes d'activité . . . . .	123

3.1.2	Évaluation du report du CSMA/CA slotté . . . . .	125
	Le modèle de Markov de CSMA/CA slotté . . . . .	125
	Évaluation du délai engendré par les mécanismes de report . . . . .	130
3.1.3	Comparaison entre le CSMA/CA slotté et le <i>polling</i> . . . . .	132
3.1.4	Amélioration des performances de la couche MAC par le découpage tem- porel . . . . .	134
3.1.5	Délai de bout-en-bout . . . . .	138
3.2	Maquettage . . . . .	140
3.2.1	Évaluation de la durée des périodes d'activité . . . . .	140
3.2.2	Délai de bout-en-bout . . . . .	141
4	Étude analytique . . . . .	143
4.1	Étude et évaluation de l'économie d'énergie . . . . .	143
4.1.1	Calcul de la durée de $[T_0; T_1]$ . . . . .	143
4.1.2	Calcul de la durée de $[T_1; T_2]$ . . . . .	143
4.1.3	Tableau récapitulatif des durées de $[T_0; T_1]$ et $[T_1; T_2]$ . . . . .	143
4.1.4	Application numérique pour les <i>scenarii</i> de la partie 3.1.4 . . . . .	146
4.2	Étude et évaluation du délai de bout-en-bout . . . . .	147
4.2.1	Trafic prioritaire montant à destination du CPAN . . . . .	148
4.2.2	Trafic prioritaire à destination quelconque . . . . .	150
4.2.3	Influence de la durée de $[T_2; T_3]$ . . . . .	151
4.2.4	Mesures par simulation du délai de bout-en-bout pour deux <i>scenarii</i> . . .	153
5	Synthèse . . . . .	157
	Les collisions : . . . . .	157
	L' <i>idle listening</i> et l' <i>overhearing</i> : . . . . .	157
	La surcharge : . . . . .	157
	Envois infructueux : . . . . .	157
	Le déterminisme : . . . . .	158
	Le délai de bout-en-bout : . . . . .	158
<b>5</b>	<b>Conclusion</b>	<b>159</b>
	<b>Nomenclature</b>	<b>162</b>
	<b>Appendices</b>	<b>167</b>
<b>A</b>	<b>Identification de la zone d'interférences</b>	<b>169</b>

## **B Format des trames MaCARI 173**

## **C Liste des primitives de MaCARI 177**

1	Primitives de données . . . . .	177
1.1	MaCARI-DATA.request . . . . .	177
1.2	MaCARI-DATA.indication . . . . .	177
1.3	MaCARI-DATA.confirm . . . . .	179
2	Primitives de contrôle . . . . .	179
2.1	MaCARI-ASSOCIATION.request . . . . .	179
2.2	MaCARI-ASSOCIATION.indication . . . . .	179
2.3	MaCARI-ASSOCIATION.confirm . . . . .	179
2.4	MaCARI-ASSOCIATION.response . . . . .	181
2.5	MaCARI-ASSOCIATION-NOTIFICATION.request . . . . .	181
2.6	MaCARI-ASSOCIATION-NOTIFICATION.indication . . . . .	181
2.7	MaCARI-DIRECT-ASSOCIATION.request . . . . .	185
2.8	MaCARI-DIRECT-ASSOCIATION.confirm . . . . .	185
2.9	MaCARI-GET.request . . . . .	185
2.10	MaCARI-GET.confirm . . . . .	186
2.11	MaCARI-SET.request . . . . .	186
2.12	MaCARI-SET.confirm . . . . .	186
2.13	MaCARI-DISASSOCIATION.request . . . . .	188
2.14	MaCARI-DISASSOCIATION.indication . . . . .	188
2.15	MaCARI-DISASSOCIATION.confirm . . . . .	188
2.16	MaCARI-DISASSOCIATION-NOTIFICATION.request . . . . .	188
2.17	MaCARI-DISASSOCIATION-NOTIFICATION.indication . . . . .	189
2.18	MaCARI-SCAN.request . . . . .	189
2.19	MaCARI-SCAN.confirm . . . . .	190
2.20	MaCARI-GTS.request . . . . .	191
2.21	MaCARI-GTS.confirm . . . . .	191
2.22	MaCARI-GTS.indication . . . . .	191
2.23	MaCARI-NEIGHBOR-COLORS.request . . . . .	192
2.24	MaCARI-NEIGHBOR-COLORS.confirm . . . . .	192
2.25	MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.request . . . . .	192
2.26	MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.confirm . . . . .	192

2.27	MaCARI-LINK-CHARACTERISTICS-LIST.request . . . . .	193
2.28	MaCARI-LINK-CHARACTERISTICS-LIST.confirm . . . . .	193
2.29	MaCARI-CONFLICT.indication . . . . .	193
2.30	MaCARI-SYNC-LOSS.indication . . . . .	194
2.31	MaCARI-LINK-LOSS.indication . . . . .	194
2.32	MaCARI-TRX-STATE.indication . . . . .	194
2.33	MaCARI-COLORING-MODE-ON.request . . . . .	195
2.34	MaCARI-COLORING-MODE-ON.confirm . . . . .	195
2.35	MaCARI-COLORING-MODE-ON.indication . . . . .	195
2.36	MaCARI-MAX-COLOR.request . . . . .	195
2.37	MaCARI-MAX-COLOR.confirm . . . . .	196
<b>Liste des publications</b>		<b>196</b>
<b>Bibliographie</b>		<b>198</b>

# Table des figures

1.1	Un exemple d'un réseau de capteurs sans fil. . . . .	26
1.2	Les ambitions d'un réseau OCARI. . . . .	33
2.1	Découpage temporel de TRAMA pour un nœud du réseau. . . . .	40
2.2	Découpage temporel de FLAMA. . . . .	41
2.3	Découpage temporel de E-MAC, L-MAC et AI-MAC. . . . .	42
2.4	Portée et zone d'interférence. . . . .	43
2.5	A et C envoient simultanément deux messages à B et génèrent un risque de collision en B. . . . .	44
2.6	A communique avec D. B veut communiquer avec C mais ne le fait pas parce que A occupe le médium, alors que si B communique avec C aucune collision n'aura lieu. . . . .	45
2.7	Séquencement des périodes d'écoute et de sommeil dans S-MAC. . . . .	46
2.8	Séquencement des périodes d'écoute et de sommeil de T-MAC et de S-MAC. . . . .	46
2.9	LPL : L'émetteur utilise un long préambule pour permettre au récepteur d'activer son module radio seulement de temps en temps. . . . .	47
2.10	Découpage temporel de Z-MAC. Notons l'existence d'un temps pendant lequel le médium n'est pas utilisé. . . . .	49
2.11	Découpage temporel de G-MAC. . . . .	50
2.12	Gestion du goulot d'étranglement dans Funneling-MAC. . . . .	51
2.13	Découpage temporel dans Funneling-MAC. . . . .	51
2.14	Exemples de topologies en étoile et en pair-à-pair. . . . .	53
2.15	La pile IEEE 802.15.4/ZigBee. . . . .	54
2.16	La structure de la supertrame. . . . .	58
2.17	Les périodes de <i>backoff</i> et les 16 slots de la supertrame. . . . .	59
2.18	Diagramme de l'algorithme de CSMA/CA slotté de la norme IEEE 802.15.4. . . . .	60
2.19	Diagramme de l'algorithme de CSMA/CA non-slotté de la norme IEEE 802.15.4. . . . .	62
2.20	Topologie <i>cluster-tree</i> . . . . .	66



2.21	Allocation des adresses hiérarchiques sur un exemple. Le coordinateur 1 par exemple, qui est à la profondeur 1, a alloué l'adresse 2 à son premier fils et 8 au deuxième, $8 = 2 + Cskip(1)$ .	68
2.22	Couche application ZigBee. L'interface entre la couche application est les différents objets est appelée APSDE pour ( <i>APplication Support sub-layer Data Entity</i> ).	69
2.23	Exemples d'application et de profils ZigBee.	70
2.24	Collision directe de <i>beacons</i> .	71
2.25	Découpage temporel de l'approche <i>Beacon Only Period</i> .	72
2.26	Découpage temporel de l'approche de supertrames asynchrones.	72
2.27	Collisions indirectes de <i>beacons</i> .	73
3.1	Exemple d'une topologie d'un réseau OCARI.	77
3.2	Le cycle global dans MaCARI.	77
3.3	Une période d'activité d'une étoile suivie par un intervalle de relais.	78
3.4	La pile OCARI.	81
3.5	Nécessité d'un coloriage à trois sauts.	82
3.6	Séquencement des slots colorés avec l'algorithme de SERENA dans $[T_2; T_3]$ .	83
3.7	Un exemple de cascade de <i>beacons</i> .	85
3.8	Cascade de <i>beacons</i> avec temps de traitement.	92
3.9	Cascade de <i>beacons</i> avec envois anticipés.	92
3.10	Exemple d'ordres de synchronisation. Un autre ordre amélioré aurait pu être : A D C B E.	94
3.11	Séquencement des périodes d'activité des étoiles.	97
3.12	Un exemple de topologie avec les différentes profondeurs.	101
3.13	Gestion des files d'attente selon le type de trafic et la destination.	103
4.1	Comparaison entre les résultats des mesures faites en intérieur et les résultats du modèle ITU-R P1238-4.	108
4.2	Diagramme de l'algorithme de routage $[T_2; T_3]$ .	114
4.3	Capture MaCAview.	115
4.4	La topologie de la maquette dans le cadre du projet OCARI.	117
4.5	Une photo des cartes utilisées pour la maquette.	117
4.6	Effet de l'envoi par anticipation sur la durée de la période de synchronisation pour des configurations de 10 à 24 coordinateurs.	119
4.7	Effet de l'envoi par anticipation sur la durée de la période de synchronisation pour des configurations de 20 à 70 coordinateurs.	120
4.8	La topologie de test de précision de la synchronisation su $T_1$ .	121

4.9	Un zoom sur les résultats de l'analyseur d'états logiques. . . . .	121
4.10	Simulation : Temps nécessaire pour écouler une certaine quantité de trafic dans une étoile en fonction du nombre de feuilles et du nombre de trames générées par feuille. . . . .	124
4.11	Simulation : Nombre de trames reçues par le coordinateur d'une étoile en fonction du nombre de feuilles et du trafic généré par chaque feuille. . . . .	124
4.12	Le modèle de Markov du CSMA/CA slotté de la norme IEEE 802.15.4 sans les mécanismes de report. . . . .	126
4.13	Chaîne de Markov représentant l'algorithme de CSMA/CA slotté de IEEE 802.15.4. . . .	127
4.14	Report de CCA à la période d'activité suivante (boîte $(k, \text{report}^C)$ de la figure 4.13). . . .	128
4.15	Report de <i>backoff</i> à la période d'activité suivante (boîte $(k, \text{report}_b^B)$ de la figure 4.13). . .	129
4.16	Attente entre la fin de la période d'activité en cours et la période d'activité suivante (boîte "attendre" des figures 4.14 et 4.15). . . . .	130
4.17	Distribution du délai d'accès au médium montrant l'effet du report. . . . .	130
4.18	Effet de la taille des trames sur le délai induit par le report. . . . .	131
4.19	Délai induit par le report dans le cas d'un réseau économe en énergie. . . . .	131
4.20	Temps de récolte avec 100 % des feuilles actives. . . . .	132
4.21	Temps de récolte avec 75 % des feuilles actives. . . . .	133
4.22	Temps de récolte avec 50 % des feuilles actives. . . . .	133
4.23	Temps de récolte avec 25 % des feuilles actives. . . . .	133
4.24	Performances de CSMA/CA slotté dans l'étoile. . . . .	134
4.25	Les trois configurations de MaCARI. . . . .	135
4.26	Nombre de collisions par trame envoyée selon la configuration pour chacun des 5 <i>scenarii</i> pour la même quantité de trafic. . . . .	136
4.27	Pourcentage de trames reçues selon la configuration pour chacun des 5 <i>scenarii</i> pour la même quantité de trafic. . . . .	137
4.28	Nombre d'octets reçus par rapport au nombre d'octets envoyés selon la configuration pour chacun des 5 <i>scenarii</i> . . . . .	137
4.29	Capture d'écran de notre animateur de simulation montrant une topologie de 6 coordina- teurs avec 1 feuille chacun et 1 CPAN. . . . .	139
4.30	Simulation : Distribution des délais de bout-en-bout pour la feuille 9 de la topologie 4.29.	139
4.31	Simulation : Distribution des délais de bout-en-bout pour la feuille 12 de la topologie 4.29.	140
4.32	Mesures : Temps nécessaire pour faire écouler une certaine quantité de trafic dans une étoile en fonction du nombre de feuilles et du nombre de trames générées par feuille. . . .	140

4.33 Mesures : Nombre de trames reçues par le coordinateur d'une étoile en fonction du nombre de feuilles et du trafic généré par chaque feuille. . . . .	141
4.34 Mesures : Distribution des délais de bout-en-bout pour la feuille 9 de la topologie 4.29. . .	141
4.35 Mesures : Distribution des délais de bout-en-bout pour la feuille 12 de la topologie 4.29. .	142
4.36 Gain énergétique avec $[T_2; T_3] = [T_1; T_2]$ et $[T_3; T_0] = [T_1; T_2] + [T_2; T_3]$ . . . . .	145
4.37 Gain énergétique avec $[T_2; T_3] = [T_1; T_2]/2$ et $[T_3; T_0] = [T_1; T_2] + [T_2; T_3]$ . . . . .	145
4.38 Abaque : gain énergétique en fonction de la durée de $[T_3; T_0]$ pour une configuration avec 8 feuilles par coordinateur. . . . .	146
4.39 Abaque : Gain énergétique pour les 5 <i>scenarii</i> . . . . .	146
4.40 Définition du pire des cas pour l'évaluation du délai de bout-en-bout pour un trafic prioritaire. . . . .	147
4.41 Taille du réseau avec $[T_2; T_3] = [T_1; T_2]$ et un trafic à destination du CPAN. . . . .	149
4.42 Taille du réseau avec $[T_2; T_3] = [T_1; T_2]$ et un trafic à destination quelconque. . . . .	151
4.43 Taille du réseau avec $[T_2; T_3] = [T_1; T_2]/2$ et un trafic à destination du CPAN. . . . .	152
4.44 Taille du réseau avec $[T_2; T_3] = [T_1; T_2]/2$ et un trafic à destination quelconque. . . . .	153
4.45 Abaque : taille du réseau en fonction de la durée de $[T_2; T_3]$ . . . . .	153
4.46 Une topologie de 5 coordinateurs avec 8 feuilles par coordinateur. . . . .	154
4.47 Distribution du délai de bout-en-bout pour la feuille 44 de la topologie 4.46. . . . .	155
4.48 Une topologie de 9 coordinateurs avec 2 feuilles par coordinateur. . . . .	155
4.49 Distribution du délai de bout-en-bout pour la feuille 26 de la topologie 4.48. . . . .	156
A.1 Position des cartes CSMA/CA et superviseur par rapport à la carte perturbatrice. . . . .	170
A.2 Nombre de trames reçues en fonction de la distance. . . . .	171
A.3 Nombre de trames perdues ou retardées envoyées par la carte CSMA/CA. . . . .	171

# Liste des tableaux

2.1	Énergie consommée par état par le composant radio d'une carte B2400ZB-tiny. . . . .	37
2.2	Récapitulatif des protocoles étudiés en terme d'économie d'énergie et de qualité de service. . . . .	52
2.3	Le débit en fonction de la fréquence et de la modulation. . . . .	55
2.4	Format du <i>beacon</i> de la norme IEEE 802.15.4. . . . .	64
2.5	Valeurs de <i>Cskip</i> pour l'exemple (3, 3, 5). . . . .	67
3.1	Format général d'une trame MaCARI. . . . .	79
3.2	Format du <i>beacon</i> MaCARI. . . . .	86
3.3	Le <i>payload</i> du <i>beacon</i> . . . . .	87
4.1	Précision sur le calcul de $T_1$ en $\mu s$ . . . . .	122
4.2	Taille des périodes $[T_0; T_1]$ et $[T_1; T_2]$ en fonction du nombre de nœuds. . . . .	144
4.3	Équations pour le trafic à destination du CPAN. . . . .	152
4.4	Équations pour le trafic à destination quelconque. . . . .	152
B.1	Format général d'une trame MaCARI. . . . .	173
B.2	Le champ frame control . . . . .	173
B.3	Les types de trames . . . . .	174
B.4	Le bit PAN ID compression . . . . .	174
B.5	Addressing modes . . . . .	175
C.1	Liste des primitives de données . . . . .	177
C.2	Champs de MaCARI-DATA.request . . . . .	178
C.3	Champs de MaCARI-DATA.indication . . . . .	180
C.4	Champs de MaCARI-DATA.confirm . . . . .	181
C.5	Liste des primitives de contrôle . . . . .	182
C.6	Champs de MaCARI-ASSOCIATION.request . . . . .	183
C.7	Champs de MaCARI-ASSOCIATION.indication . . . . .	183

C.8 Champs de MaCARI-ASSOCIATION.confirm . . . . .	183
C.9 Champs de MaCARI-ASSOCIATION.response . . . . .	184
C.10 Champs de MaCARI-ASSOCIATION-NOTIFICATION.request . . . . .	184
C.11 Champs de MaCARI-ASSOCIATION-NOTIFICATION.indication . . . . .	184
C.12 Champs de MaCARI-DIRECT-ASSOCIATION.request . . . . .	185
C.13 Champs de MaCARI-DIRECT-ASSOCIATION.confirm . . . . .	185
C.14 Champs de MaCARI-GET.request . . . . .	186
C.15 Champs de MaCARI-GET.confirm . . . . .	186
C.16 Champs de MaCARI-SET.request . . . . .	186
C.17 Champs de MaCARI-SET.confirm . . . . .	187
C.18 Attributs de MaCARI (à compléter) . . . . .	187
C.19 Champs de MaCARI-DISASSOCIATION.request . . . . .	188
C.20 Champs de MaCARI-DISASSOCIATION.indication . . . . .	188
C.21 Champs de MaCARI-DISASSOCIATION.confirm . . . . .	189
C.22 Champs de MaCARI-DISASSOCIATION-NOTIFICATION.request . . . . .	189
C.23 Champs de MaCARI-DISASSOCIATION-NOTIFICATION.indication . . . . .	189
C.24 Champs de MaCARI-SCAN.request . . . . .	190
C.25 Champs de MaCARI-SCAN.confirm . . . . .	190
C.26 Champs de MaCARI-GTS.request . . . . .	191
C.27 Champs de MaCARI-GTS.confirm . . . . .	191
C.28 Champs de MaCARI-GTS.indication . . . . .	192
C.29 Champs de MaCARI-NEIGHBOR-COLORS.request . . . . .	192
C.30 Champs de MaCARI-NEIGHBOR-COLORS.confirm . . . . .	192
C.31 Champs de MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.confirm . . . . .	193
C.32 Champs de MaCARI-LINK-CHARACTERISTICS-LIST.confirm . . . . .	193
C.33 Champs de MaCARI-CONFLICT.indication . . . . .	194
C.34 Champs de MaCARI-SYNC-LOSS.indication . . . . .	194
C.35 Champs de MaCARI-LINK-LOSS.indication . . . . .	194
C.36 Champs de MaCARI-TRX-STATE.indication . . . . .	195
C.37 Champs de MaCARI-COLORING-MODE-ON.confirm . . . . .	195
C.38 Champs de MaCARI-MAX-COLOR.request . . . . .	196
C.39 Champs de MaCARI-MAX-COLOR.confirm . . . . .	196

# Introduction

Les avancées technologiques du monde de l'électronique ont permis la conception de nœuds communicants miniatures capables de fonctionner avec peu d'énergie. Leur petite taille souvent de l'ordre de celle d'une pièce de monnaie et leur sobriété énergétique limitent leurs capacités de calcul et de mémoire. Un réseau de capteurs sans fil peut être défini comme un ensemble de tels nœuds, chacun couplé à un capteur ou à un actionneur, interagissant avec l'environnement réel. Ces nœuds participent à la transmission de données ou d'ordres et consignes, via le médium sans fil, vers une ou plusieurs unités de collecte ou de contrôle. Grâce à leur fort potentiel, les réseaux de capteurs se retrouvent au cœur de nombreuses applications couvrant des domaines aussi variés que la sécurité, l'environnement, la médecine, l'écologie, le militaire et l'industrie.

Pour concevoir une solution efficace, il faut lever bon nombre de verrous technologiques. Les solutions implémentées disponibles actuellement pour les réseaux sans fil ne répondent pas encore complètement aux spécificités des réseaux de capteurs ni à leurs applications. Les besoins et contraintes attachés aux réseaux de capteurs sans fil diffèrent de façon significative de ceux et celles des réseaux locaux sans fil. Des paramètres comme l'économie énergétique et l'encombrement des nœuds font qu'il s'agit de fait de deux domaines mitoyens et complémentaires dans le monde des réseaux.

Contrairement aux réseaux traditionnels, l'enjeu principal n'est pas l'optimisation des performances des protocoles du réseau. Les objectifs essentiels des réseaux de capteurs sans fil sont plutôt l'économie d'énergie en ayant pour but de prolonger la durée de vie du réseau, et le délai de transmission et de récupération de l'information afin de garantir un niveau acceptable aux performances du réseau. À ces besoins se rajoutent d'autres contraintes comme la robustesse, l'auto-configuration et la mobilité des entités qui sont liées à l'application ainsi qu'à l'environnement dans lequel le réseau de capteurs est déployé.

Pour répondre à ces besoins, les protocoles employés dans les réseaux de capteurs sans fil doivent être capables d'assurer leurs fonctionnalités principales tout en prenant ces contraintes en compte. Par exemple, un protocole de routage qui a pour rôle de trouver un chemin vers une destination donnée doit réaliser ceci tout en préservant l'énergie des nœuds impliqués par le choix du chemin. De même, la qualité

de service doit être pensée et assurée à tous les niveaux de la pile réseau. Bon nombre d'applications génèrent un type de trafic qui nécessite des performances particulières comme un temps d'accès au médium borné. Au niveau de la couche MAC, la qualité de service se traduit généralement par la mise en œuvre de mécanismes d'accès au médium qui garantissent l'envoi d'une trame sans collision dans un délai borné. Dans ce cas, nous parlons de l'aspect déterministe d'une couche MAC pour un trafic en temps contraint.

Dans le cadre de cette thèse, nous allons travailler sur une conception d'une couche MAC déterministe pour les réseaux de capteurs sans fil économes en énergie. Garantir à la fois l'économie d'énergie et le déterminisme est un défi où un juste compromis entre ces deux aspects doit être réalisé. Ils sont en effet antagonistes. Pour économiser de l'énergie, une entité doit se mettre en mode sommeil, mode durant lequel elle sera inactive. Étant inactive, une entité n'est plus capable de participer à l'activité du réseau ; elle ne peut ni recevoir ni transmettre des messages, qu'il s'agisse de traiter les informations qui la concerne ou de participer au routage en tant que relais. Ainsi, le fait de se mettre en état inactif est un handicap pour l'aspect déterministe et pour le respect de contraintes temporelles fortes.

Ce travail a été réalisé au sein de l'équipe Réseaux et Protocoles du laboratoire LIMOS de l'Université Blaise Pascal de Clermont-Ferrand. Il s'intègre dans le cadre du projet OCARI. Le projet OCARI (Optimisation des Communications Ad hoc pour les Réseaux Industriels) est un projet de recherche de type intégration financé par l'ANR (Agence Nationale de Recherche). Ce projet vise à exploiter les avancées technologiques dans le domaine des réseaux ad hoc pour ouvrir le champ aux usages des technologies sans fil en milieu industriel. Il a donc pour ambition de participer au développement d'une nouvelle génération de réseaux basée sur l'emploi des réseaux de capteurs sans fil en environnement industriel, en prenant en compte les environnements fortement contraints et permettant de favoriser de nouvelles applications industrielles. Notre contribution dans ce projet est centrée au niveau de la conception et du développement d'une méthode d'accès adaptée au déterminisme et aux exigences de qualité de service telles que l'absence de collision et la garantie d'un délai borné de bout-en-bout tout en conservant une faible consommation énergétique.

Dans le premier chapitre, nous commençons par une introduction détaillée des réseaux de capteurs sans fil en insistant sur leurs spécificités et en citant quelques exemples de domaines d'application. Nous terminons le premier chapitre en situant le travail effectué durant cette thèse dans le contexte général du projet OCARI.

Dans le deuxième chapitre, nous enchaînons avec une étude des protocoles d'accès au médium proposés pour les réseaux de capteurs sans fil. Nous insistons sur les différentes techniques utilisées pour répondre aux besoins de ce type de réseaux. Nous étudions aussi la norme IEEE 802.15.4 et le standard Zigbee, ce dernier offrant une pile de réseau complète dédiée aux réseaux de capteurs sans fil.

Dans le troisième chapitre, nous proposons le protocole MaCARI (MAC pour OCARI), un protocole d'accès au médium économe en énergie et déterministe. MaCARI adopte une segmentation temporelle qui améliore les performances du réseau et permet de garantir l'accès au médium pour un trafic de type prioritaire. Le temps dans MaCARI est découpé en cycles globaux. Chaque cycle global est segmenté temporellement en plusieurs périodes séquentielles pour améliorer l'utilisation du médium et permettre à toutes les entités du réseau d'économiser de l'énergie.

Ensuite, nous présentons dans le quatrième chapitre notre démarche de validation des mécanismes de MaCARI, ceci par une étude analytique, par des simulations et par des mesures sur une maquette. Les résultats concernant différentes configurations montrent que MaCARI améliore d'une façon très significative l'utilisation du médium et garantit une économie d'énergie très importante pour l'ensemble du réseau.

Nous concluons notre travail dans le cinquième chapitre en discutant des différentes perspectives offertes par ce sujet.





# Chapitre 1

## Les réseaux de capteurs sans fil

Un réseau de capteurs sans fil est un ensemble de petites entités à faible coût, appelées nœuds capteurs, limitées en ressources énergétiques et en capacité de calcul, et qui communiquent entre elles via des antennes et la modulation d'une onde radio. Chaque entité est composée d'un capteur ou d'un actionneur et d'une carte réseau sans fil. Le capteur (ou l'actionneur) est dédié à une tâche bien précise comme un relevé de température, une détection de présence ou la commande d'une électrovanne. L'information produite par le capteur est ensuite passée à la carte réseau du nœud qui la communique généralement via le réseau à une unité de surveillance ou de contrôle.

Le déploiement des capteurs dépend de l'application. Il peut être complètement aléatoire comme dans le cas de la surveillance de sites naturels où le nombre de capteurs est de l'ordre de quelques milliers. Pour d'autres applications, un déploiement peut être bien précis, c'est le cas du suivi d'une activité industrielle où le nombre de capteurs est plutôt de l'ordre de quelques centaines.

Le fait que la communication soit sans fil permet d'avoir plus de flexibilité pour le déploiement par rapport à un déploiement filaire qui demande du câblage. Dans certaines applications comme la surveillance d'une application à bord d'une frégate, le poids des câbles utilisés devient une surcharge conséquente.

Pour les applications nécessitant la mobilité, il ne faut pas remplacer la contrainte du câblage par une solution encombrante ou amenant de l'inertie. Un nœud doit donc résulter d'une volonté d'intégration. Il est courant de comparer le volume qui devrait être occupé par un nœud à celui d'une pièce de monnaie. Les solutions disponibles à ce jour sont plutôt construites à partir d'un micro-contrôleur relié à une composante spécifique radio. Cet assemblage est un compromis caractérisé par une intégration insuffisante et une capacité limitée de calcul et de mémoire qui se révèle être une contrainte lorsque les protocoles deviennent de plus en plus complexes. Si de plus la solution doit être autonome énergétiquement, cela a un impact à la fois sur le coût mais aussi sur l'encombrement.

Les protocoles réseaux utilisés doivent prendre en compte cette limitation énergétique. En effet remplacer une entité du réseau ou renouveler ses ressources énergétiques après déploiement peut devenir très coûteux (notamment lorsqu'il s'agit de systèmes embarqués qui demandent d'arrêter le système pour permettre cette intervention) et parfois impossible selon la nature du déploiement (capteurs installés dans des zones de radioactivité par exemple).

La portée des antennes de communication est aussi affectée par cette contrainte énergétique et l'étendue d'un réseau de capteurs dépasse souvent largement la portée des antennes utilisées. Cela force les nœuds capteurs à échanger des messages multi-sauts pour atteindre la destination. La figure 1.1 montre un exemple d'un réseau de capteurs où les nœuds communiquent de proche en proche pour atteindre le nœud passerelle qui à son tour communique avec une unité de contrôle. Dans cet exemple le réseau de capteurs est connecté à une station de contrôle via une passerelle en passant par un autre réseau de technologie différente (filaire ou sans fil). La liaison entre le réseau de capteurs et la passerelle peut être filaire ou sans fil selon les besoins.

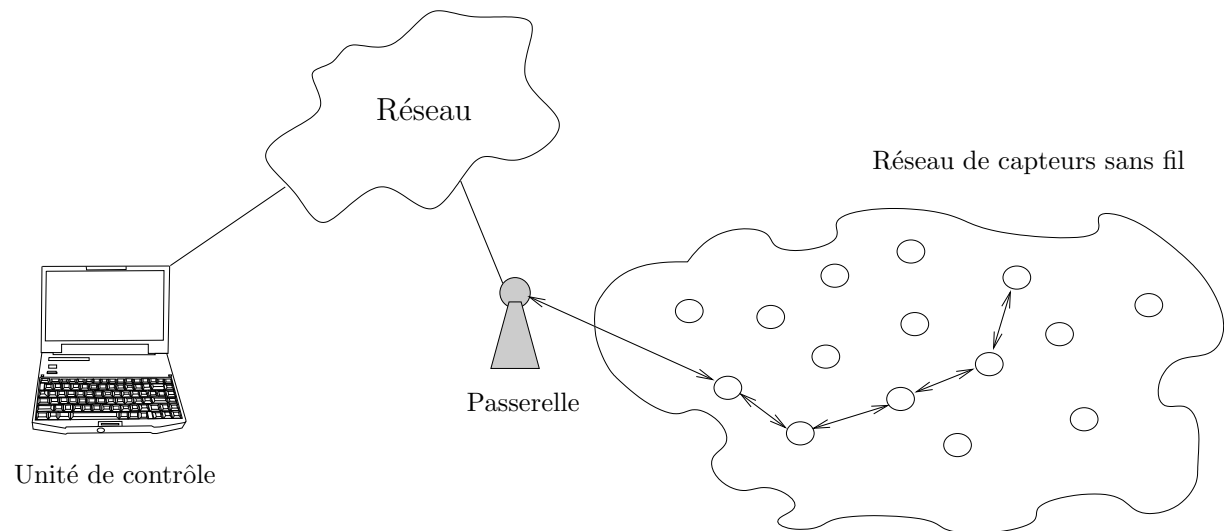


FIG. 1.1 – Un exemple d'un réseau de capteurs sans fil.

Même en diminuant la puissance de transmission, l'interface radio reste l'équipement le plus consommateur en énergie dans un nœud capteur. Comme la couche MAC contrôle l'accès au médium c'est-à-dire l'utilisation de cette interface pour transmettre les messages, c'est le protocole MAC qui est le plus concerné par la consommation énergétique dans un réseau de capteurs sans fil.

Dans ce cadre, nous allons nous centrer sur les protocoles MAC des réseaux de capteurs sans fil en étudiant les différentes techniques utilisées dans ce domaine afin d'optimiser la consommation énergétique. Nous proposons dans la suite le protocole MAC nommé MaCARI qui répond à certains besoins des applications industrielles des réseaux de capteurs sans fil.

## **1 Domaines d'application des réseaux de capteurs sans fil**

La taille de plus en plus réduite des micro-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, vibrations, ...) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'applications [1]. Ils permettent aussi d'étendre les applications existantes et de faciliter la conception d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Les réseaux de capteurs peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, nous pouvons citer les domaines suivants : militaire, environnemental ou domestique, mais aussi de la santé, de la sécurité, de l'écologie, de traçabilité, etc. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous.

### **1.1 Applications militaires**

Le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs, comme c'est le cas pour plusieurs technologies. Le déploiement rapide, l'auto-configuration et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine [2]. Un exemple typique d'application dans ce domaine est le déploiement d'un tel réseau sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies [3] ou d'analyser le terrain avant d'y envoyer des troupes (par la détection d'agents chimiques, biologiques ou de radiations, par exemple).

### **1.2 Applications liées à la sécurité**

L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant des résultats plus fiables. Comme exemple des applications de ce type nous pouvons citer : (i) la détection des altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, par des capteurs intégrés dans les murs ou dans le béton, (ii) la surveillance des mouvements afin de constituer un système de détection d'intrusions distribué. L'aspect distribué rend plus complexe la possibilité de mettre hors d'usage ce système de surveillance.

### **1.3 Applications environnementales**

La surveillance de larges sites naturels ou industriels utilise les réseaux de capteurs qui garantissent l'aspect distribué et une large couverture [4]. Dans ce domaine nous pouvons mentionner des applications

typiques : (i) la dispersion de thermo-capteurs à partir d'un avion sur une forêt pour signaler un éventuel début d'incendie dans le champ de captage, (ii) le semis de nœuds capteurs en même temps avec les graines dans les champs agricoles pour pouvoir identifier les zones sèches et rendre l'irrigation plus efficace, (iii) le déploiement de nœuds capteurs sur les sites industriels, les centrales nucléaires ou dans les raffineries de pétrole pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace.

## **1.4 Applications médicales**

L'utilisation des réseaux de capteurs dans le domaine de la médecine peut apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité [5], facilitant ainsi le diagnostic de maladies grâce à des micro-capteurs qui pourront être ingérés ou implantés sous la peau. Comme applications d'avant garde de ce domaine nous pouvons énumérer : (i) les micro-caméras qui peuvent être ingérées et sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain, (ii) la création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision de l'œil.

## **1.5 Applications écologiques**

Dans un contexte mondial où le réchauffement de la planète devient une préoccupation grandissante, l'utilisation des réseaux de capteurs pour optimiser la consommation des ressources énergétique peut avoir une conséquence environnementale positive [6]. Un exemple de ce type d'application est l'intégration de plusieurs micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenchés qu'aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué peut aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait probablement de réduire la demande mondiale en énergie.

## **1.6 Applications de traçabilité et de localisation**

Suite à une avalanche il est nécessaire de localiser les victimes enterrées sous la neige en équipant les personnes susceptibles de se trouver dans des zones à risque par des capteurs. Ainsi, les équipes de sauvetage peuvent localiser plus facilement les victimes [7]. Contrairement aux solutions de traçabilité et de localisation basées sur le système de GPS (*Global Positionning System*), les réseaux de capteurs peuvent être très utiles dans des endroits clos comme les mines par exemple [8].

## 1.7 Applications industrielles

Si nous considérons maintenant les applications qualifiées "d'industrielles" pour désigner des activités touchant à la production de produits manufacturés ou des fournisseurs d'énergie, dans ce domaine, les réseaux de capteurs sans fil offrent une grande flexibilité d'emploi puisqu'ils permettent de s'affranchir des contraintes liées aux câblages. Il est alors possible de satisfaire des contraintes de poids (d'un équipement dans un navire), de mobilité (d'un robot dans une usine), de facilité de déploiement (sur un site préexistant), d'isolation galvanique... Parmi ces applications nous pouvons citer comme exemples : (i) la surveillance des ouvrages de génie civil pour établir un diagnostic à distance d'une évolution de leur état sans avoir à se rendre sur place pour faire une observation *in situ*, (ii) la surveillance de l'état de santé d'un ouvrier ou du risque de le voir exposé à des conditions de travail dangereuses (exposition à la radioactivité), (iii) l'observation d'un site susceptible de subir les effets d'une pollution et la construction en temps réel d'une cartographie de sa contamination grâce à des capteurs, disséminés sur le site, capables de relever les niveaux de pollution et de relayer l'information vers les services spécialisés.

## 2 Besoins des applications industrielles

Les réseaux de capteurs sans fil sont de plus en plus utilisés pour contrôler et surveiller des applications industrielles. Ces applications sont caractérisées par un certain nombre de besoins et génèrent des contraintes que le réseau doit prendre en compte. Ces besoins sont liés à leur environnement et à la nature de leur activité [9]. Nous avons identifié cinq besoins essentiels : l'économie d'énergie, la qualité de service, l'auto-configuration, la mobilité et la sécurité. Par la suite, nous allons mettre en avant les besoins et les contraintes des applications industrielles de manière générale.

### 2.1 Économie d'énergie

La nature de l'environnement industriel exige parfois un déploiement des entités du réseau dans des endroits confinés et difficiles d'accès pour les êtres humains : des champs radioactifs, des barrages, des systèmes embarqués, etc. Ces entités sont souvent auto-alimentées par des piles et remplacer une pile dans ces conditions de déploiement n'est pas toujours possible. Une des contraintes industrielles est l'aspect économe en énergie qui augmente la durée de vie du réseau et évite de remplacer les entités, ou leurs piles, trop souvent.

## **2.2 Qualité de service**

Dans les applications industrielles, le trafic échangé sur le réseau peut être constitué d'informations critiques pour le bon déroulement de l'activité industrielle. Certaines informations possèdent une courte durée de validité et deviennent inutiles au bout d'un certain temps. De telles informations doivent être acheminées aux entités concernées en prenant en considération ces contraintes temporelles. Les applications industrielles exigent une forte contrainte de qualité de service en général et de déterminisme en particulier au niveau de la couche MAC pour garantir la récupération des informations essentielles et garantir la validité temporelle de ces informations.

## **2.3 Auto-configuration**

Les réseaux de capteurs sans fil sont déployés dans des entreprises pour lesquelles les réseaux sans fil ne font pas partie du cœur de métier. Dans ce cas, il n'est pas toujours opportun que de telles entreprises entretiennent une expertise réseau. Cette contrainte force le réseau à être auto-configurable. L'ajout d'une nouvelle entité au réseau doit se faire sans l'intervention d'un expert réseau. L'auto-configuration répond aussi au besoin de l'adaptation aux conditions d'évolution de l'environnement. Une entité qui se retrouve déconnectée du réseau doit pouvoir se reconnecter sans une intervention externe, à défaut, le routage doit compenser la perte de cette station.

## **2.4 Mobilité**

La prise en compte de l'aspect mobilité est un besoin qui dépend de la nature de l'application. Certaines applications industrielles demandent des interventions du personnel qui forcent celui-ci à se déplacer souvent pour effectuer des tâches comme le relevé d'informations ou la mise à jour d'un appareil. L'utilisation du sans fil dans le monde de l'industrie ne se limite pas à la mobilité des utilisateurs, mais elle devient une nécessité quand le fonctionnement propre de certaines machines constituées de parties mobiles empêchent l'utilisation de solutions câblées.

## **2.5 Sécurité**

Les informations échangées par un réseau de capteurs sans fil sont diffusées dans l'air sous la forme d'une modulation radio. Elles sont donc exposées à une écoute non autorisée et à des altérations intentionnelles ou non. La protection des informations échangées et la sécurité contre les attaques malicieuses qui risquent de perturber l'activité du réseau est un souci exprimé par les applications qui nécessitent d'avoir un réseau protégé et fiable. Une attaque sur le réseau peut mettre en danger l'activité de l'industrie et causer des interruptions de production.

### 3 Projet OCARI

Le travail effectué durant cette thèse entre dans le cadre du projet OCARI (Optimisation des Communications Ad-hoc pour les Réseaux Industriels) [10] financé par l'ANR (Agence Nationale de Recherche) [11]. Ce projet a pour ambition de participer au développement d'une nouvelle génération de réseau sans fil de type ad-hoc dédié à un environnement industriel. Il s'agit de réseaux sans fil robustes, prenant en compte les environnements fortement contraints et permettant de répondre aux besoins d'applications industrielles. L'objectif du projet est la conception d'un pile réseau où chaque couche tend à répondre à ces besoins.

#### 3.1 Les partenaires du projet OCARI

Le projet OCARI englobe trois partenaires industriels (DCNS, EDF et Telit) et quatre partenaires académiques (INRIA, LATTIS, LIMOS et LRI). Nous décrivons brièvement le rôle de chacun des partenaires :

- EDF (Électricité De France) est le chef du projet. Ce partenaire a des besoins industriels qu'il compte satisfaire en utilisant les technologies sans fil des réseaux de capteurs. Ces applications couvrent la surveillance des barrages, l'équipement des sites nucléaires jusqu'aux applications de dosimétrie.
- La DCNS (Direction de Constructions Navales) travaille conjointement avec EDF sur la conception du *middleware* de la pile OCARI. Ce partenaire a des besoins de mise en place de technologies sans fil à bord de fré gates ou d'autres navires dans des environnements très confinés et contraints.
- L'INRIA (Institut National de Recherche en Informatique et Automatique) a comme rôle de proposer des protocoles de niveau 3 qui répondent au besoin du projet OCARI.
- Le LRI (Laboratoire de Recherche en Informatique) évalue l'efficacité des protocoles niveau MAC et réseau de la pile OCARI en termes d'économie d'énergie.
- Le LATTIS (LABoratoire Toulousain de Technologie et d'Ingénierie des Systèmes) participe avec le LIMOS à la conception et au maquetage de la couche MAC de la pile OCARI.
- Le LIMOS (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes) travaille conjointement avec le LATTIS sur la couche MAC de la pile OCARI.
- Telit est le partenaire qui prend en charge l'intégration des différents codes produits par les partenaires du projet et fournit les cartes sur lesquelles sont développés les maquettes et les prototypes.



## 3.2 Objectifs du projet OCARI

Le projet OCARI vise à couvrir des applications variées avec des exigences à plusieurs niveaux. Nous pouvons en citer quelques unes : un délai borné de transit de l'information pour une catégorie de trafic au moins, une basse consommation énergétique pour les nœuds alimentés par des piles, le déterminisme et la fiabilité des communications.

L'objectif étant de concevoir des protocoles de réseaux qui satisfont ces besoins essentiels, ces protocoles doivent garantir à la fois un fonctionnement économe en énergie pour prolonger la durée de vie du réseau, et une qualité de service qui répond aux exigences des différentes applications. Dans ce cadre, nous travaillons sur la conception et l'évaluation d'une méthode d'accès qui prend en compte ces deux aspects.

Les réseaux OCARI sont composés d'îlots. Chaque îlot est constitué de capteurs et actionneurs formant un réseau de capteurs sans fil. Les îlots sont interconnectés entre eux via une infrastructure qui peut être filaire. Dans chaque îlot, une entité spécifique joue le rôle d'une passerelle entre le réseau de capteurs sans fil et l'infrastructure. Cette passerelle intègre un *middleware* qui a comme rôle d'harmoniser les informations récoltées du réseau de capteurs d'une part, et de surveiller et de mettre à jour les paramètres de configuration de ce réseau d'une autre part. Dans OCARI, les îlots sont séparés physiquement (géographiquement ou par le biais de canaux disjoints), la connectivité entre eux est assurée via les passerelles ou bien grâce à une entité mobile appelée rondier. Le rôle du rondier est de faire des interventions localisées et transporter des informations d'un îlot à un autre.

Dans le projet OCARI, deux types de production de données ont été identifiés :

- des données générées périodiquement à des fréquences basses (typiquement : périodes de 1 ou de plusieurs secondes) et destinées à être transmises au plus tôt à une unité de supervision,
- des données générées périodiquement stockées localement dans le nœud ou dans le réseau en attente de passage d'une entité mobile chargée de les collecter (c'est le rôle du rondier).

Dans le cadre de ce travail, nous nous consacrons au premier type de production.

La figure 1.2 montre un exemple d'un réseau OCARI avec trois îlots interconnectés via des passerelles à travers une infrastructure donnée. Une unité de contrôle est connectée aux passerelles à travers la même infrastructure. Le rôle de cette unité de contrôle est de surveiller l'activité industrielle et mettre à jour les paramètres du réseau. Nous pouvons aussi voir un rondier qui se déplace entre les îlots, se connecte à un îlot à travers une liaison sans fil avec les entités de cet îlot. La taille des topologies des réseaux OCARI est de l'ordre de 400 nœuds ou entités par îlot. La mobilité est restreinte à la vitesse du déplacement d'un rondier qui récolte des données de temps en temps.

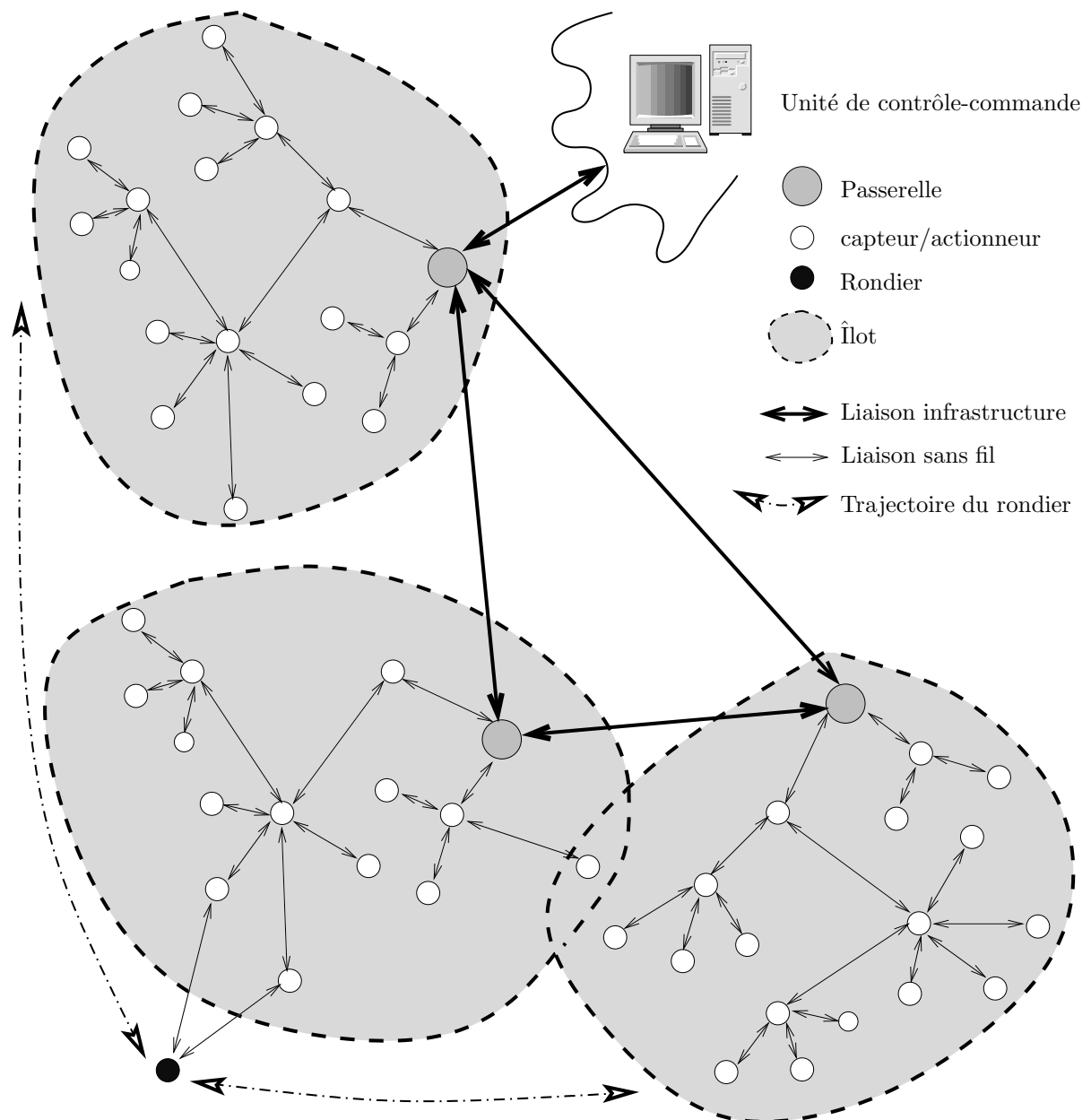


FIG. 1.2 – Les ambitions d'un réseau OCARI.



## Chapitre 2

# État de l’art des protocoles MAC pour les réseaux de capteurs sans fil

Ce chapitre est découpé en deux parties. Dans la première partie, nous présentons les protocoles MAC les plus cités dans la littérature proposés pour les réseaux de capteurs sans fil. Nous découpons ces protocoles en trois familles : les protocoles qui sont basés sur un accès aléatoire en CSMA/CA, les protocoles basés sur un accès séquentiel selon un découpage temporel TDMA et les protocoles hybrides qui mélangent ces deux solutions. Dans la seconde partie, nous détaillons la norme IEEE 802.15.4 et la norme ZigBee qui constituent les deux principaux standards actuels proposés pour les réseaux de capteurs sans fil. Nous expliquons les différentes fonctionnalités de ces deux normes en insistant sur la couche MAC et la couche réseau.

### 1 Les méthodes d’accès dans les réseaux de capteurs sans fil

Dans cette partie, nous allons détailler les méthodes d’accès des réseaux de capteurs sans fil les plus citées dans la littérature, en nous focalisant sur les contraintes spécifiques de ce type de réseau. Ces contraintes rendent les méthodes d’accès sans fil classiques, comme celles de la norme IEEE 802.11, inadaptées et inefficaces. En partant des contraintes, nous expliquons comment différents protocoles MAC ont été proposés pour prendre en considération les limitations des réseaux de capteurs.

#### 1.1 Contraintes et spécificités des réseaux de capteurs sans fil

La contrainte la plus forte des réseaux de capteurs sans fil est la limitation en ressources, que ce soit en terme de capacité de calcul, d’espace mémoire ou de ressources énergétiques. Cette contrainte est souvent

peu dominante voire ignorée par les protocoles MAC sans fil traditionnels qui essaient d'augmenter le débit, de minimiser la latence et d'être robuste par rapport à la mobilité.

Le moyen le plus efficace pour prolonger la vie d'un nœud capteur est de le mettre en mode sommeil le plus souvent possible. Cependant, un nœud en mode sommeil est un nœud inactif qui ne peut ni communiquer ni assurer son rôle dans le réseau. Cela induit des effets négatifs sur les performances du réseau. L'enjeu est donc de trouver un compromis entre la durée de vie d'un réseau et ses performances. Nous définissons la durée de vie du réseau en considérant le temps pendant lequel ce réseau satisfait aux exigences des applications en terme de qualité de service. On appelle cycle d'activité d'un nœud ou d'un réseau le rapport entre le temps passé en mode actif et le temps passé en mode sommeil.

Le cycle d'activité d'un réseau est fortement lié à l'application et à ses besoins en terme de délai, de latence et de réactivité. En général, les réseaux de capteurs sans fil sont conçus pour des applications qui diffèrent de celles des réseaux sans fil traditionnels. Les applications des réseaux de capteurs sont typiquement le monitoring, la surveillance et la traçabilité. Les caractéristiques du trafic généré par chaque application diffèrent largement surtout en terme de quantité et de fréquence.

Les réseaux de capteurs sans fil partagent avec les réseaux sans fil traditionnels les problématiques liés à l'aspect multi-sauts des réseaux sans fil ad hoc (comme le problème du terminal caché et du terminal exposé [12], les collisions, le partage du médium sans fil, etc). À noter que le problème essentiel dans les réseaux ad hoc est la mobilité, alors que dans les réseaux de capteurs le degré de mobilité est lié à l'application.

Le challenge est de pouvoir trouver des solutions à ces problèmes tout en respectant les besoins et les contraintes des réseaux de capteurs sans fil et en gardant toujours des performances réseaux globales satisfaisantes pour le type d'applications visées.

La limitation en capacité de calcul et en espace mémoire limite les fonctionnalités qu'un protocole MAC peut fournir. Souvent les protocoles MAC proposés sous-estiment le temps et la quantité de mémoire nécessaires pour exécuter leurs algorithmes. Un protocole MAC complexe qui s'adapte aux changements des conditions du réseau peut optimiser la consommation énergétique mais risque de ne pas être implémentable avec les capacités faibles des microcontrôleurs utilisés dans la fabrication des nœuds capteurs. À noter que le protocole MAC partage les ressources de calcul et de mémoire avec les autres protocoles de la pile réseau.

La nature des applications des réseaux de capteurs exige non seulement la prise en compte de l'économie d'énergie, mais aussi le respect d'une qualité de service en terme de fonctionnement temps réel. Les données générées par les capteurs doivent être acheminées vers leur destination (le plus souvent une station de contrôle), pour être traitées et analysées selon certaines contraintes temporelles. L'environnement du déploiement d'un réseau de capteurs demande une capacité d'auto-configuration pour

assurer une facilité de mise en place surtout pour des topologies de quelques centaines de nœuds.

Par la suite nous allons détailler le rôle de la couche MAC et son effet sur chacune des contraintes identifiées.

### 1.1.1 Économie d'énergie

Les sources de consommation d'énergie sur un nœud capteur sont le module radio, le microprocesseur et le capteur. La communication radio est souvent la plus consommatrice parmi les trois. La consommation d'un capteur varie dans une très large plage selon son type. Un capteur consommant beaucoup d'énergie est souvent alimenté par sa propre source énergétique. Nous considérons ici que la couche MAC est concernée uniquement par l'utilisation du module radio et du microprocesseur.

Dans un réseau de capteurs, la portée est de l'ordre d'une dizaine de mètres dans un milieu clos avec une puissance d'émission de 0 dBm (1 mW). Avec ce niveau de puissance d'émission, l'énergie consommée pour la réception et celle consommée pour l'émission sont quasiment égales. Le tableau 2.1 montre un exemple d'énergie consommée pour chaque état du module radio d'une carte B2400ZB-tiny [13].

État	Énergie consommée
Émission	26 mA
Réception ou écoute	29 mA
Veille	15 $\mu$ A
Sommeil	3 $\mu$ A

TAB. 2.1 – Énergie consommée par état par le composant radio d'une carte B2400ZB-tiny.

Un protocole MAC économe en énergie essaie d'utiliser le moins souvent possible le module radio. Les modules radio peuvent avoir plusieurs niveaux de consommation quand ils ne sont pas en mode émission ou réception, moins le nœud consomme moins il est réactif, c'est pour cela que les différents états de sommeil existent pour assurer une flexibilité selon le degré de réactivité demandé par la couche MAC.

L'utilisation inutile du module provient de 5 sources essentielles : le *Overhearing*, les collisions, le *Idle listening*, les envois infructueux et les messages de contrôle.

**Overhearing** L'*overhearing* est la réception par un nœud d'une trame qui ne lui est pas destinée. L'énergie consommée pour la réception et le traitement des données de cette trame est perdue et sans aucun intérêt.

**Collisions** Les collisions sont à la fois une source de dégradation des performances du réseau et de perte d'énergie. Les pertes de trames à cause des collisions forcent les nœuds à retransmettre le même paquet plusieurs fois et donc à rester actif pour le répéter et vérifier qu'il est bien reçu par la destination. À noter que les retransmissions ne se font que pour les trames envoyées en mode *unicast* (à un seul destinataire) et avec une demande d'acquittement.

**Idle listening** C'est l'attente d'une trame par le module radio. Cela arrive quand il a été demandé à un nœud d'être éveillé mais qu'il ne reçoit aucune trame et n'en transmet aucune non plus. Même si le nœud ne transmet pas et ne reçoit pas, le fait que son module radio soit activé et prêt pour recevoir consomme autant d'énergie que pour la réception.

**Envois infructueux** Cela arrive quand un nœud essaie de communiquer avec un autre nœud qui n'est plus accessible parce qu'il est en mode sommeil par exemple (ou hors de portée). Le nœud émetteur est en attente d'un acquittement, et il retransmet donc la même trame plusieurs fois. Il consomme de l'énergie en le faisant du fait qu'il soit resté en mode transmission et en mode réception pour l'éventuel acquittement.

**Overhead** L'*Overhead* constitue les messages de contrôle et tout bit utilisé pour encapsuler des données utiles à l'application. Cet *Overhead* généré par la couche MAC, a pour but d'assurer le bon fonctionnement de la méthode d'accès mais cela amplifie les 4 sources de consommation à contre-temps.

### 1.1.2 Déterminisme

L'aspect déterminisme est une autre caractéristique essentielle liée aux contraintes applicatives. Assurer un fonctionnement déterministe dans un réseau de capteurs sans fil consiste à pouvoir acheminer un paquet de données d'une source vers une destination finale, dans un délai borné avec une certaine probabilité de réussite (99 % par exemple). Pour ce faire, la couche MAC doit garantir un délai déterminé pour accéder au médium sans risque de collision.

Les protocoles MAC basés sur CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) ne garantissent ni un délai borné d'accès au médium, ni une transmission sans collision et par conséquent n'offrent pas un délai borné de bout en bout dès que la charge du réseau augmente. En revanche, les protocoles MAC basés sur TDMA (*Time Division Multiple Access*) , avec un bon dimensionnement des intervalles de temps alloués selon le trafic généré et son orientation, peuvent assurer d'une manière déterministe un délai borné de bout en bout dans une topologie multi-sauts. Plus de détails seront donnés en 1.2.

### 1.1.3 Auto-configuration

La nature des applications, notamment les applications militaires, des réseaux de capteurs sans fil peut nécessiter une création de réseau dynamique. Cela induit que la couche MAC doit supporter des topologies larges, denses et dynamiques. L'aspect dynamique provient de la mobilité, la défaillance occasionnelle de certains nœuds, de l'ajout de nœuds ou des changements de conditions de propagation.

## 1.2 Exemples de protocoles MAC pour les réseaux de capteurs sans fil

Un protocole MAC pour les réseaux de capteurs sans fil doit trouver le bon compromis entre l'économie d'énergie, l'aspect temps réel et l'auto-configuration. Dans cette partie, nous présentons quelques protocoles MAC proposés pour les réseaux de capteurs sans fil présentés selon leur type. Nous avons identifié trois types : les protocoles basés sur un séquençement temporel, les protocoles basés sur un évitement de collision probabiliste et les protocoles hybrides. Nous décrivons en résumé quelques protocoles MAC représentatifs de chacun de ces types. À noter que ces protocoles sont les plus étudiés dans la littérature et non pas forcément les plus performants : il s'agit de protocoles de base et parfois sujets à optimisation.

### 1.2.1 Protocoles basés sur un séquençement temporel

Les protocoles MAC appliquant un séquençement temporel découpent le temps en *slots*, chaque slot de temps est alloué (localement) à un seul nœud pour lui garantir l'accès au canal afin qu'il puisse transmettre ses données. La méthode TDMA est un exemple de méthode de séquençement temporel dans les réseaux sans fil.

Ces protocoles sont généralement économes en énergie car ils évitent les collisions, ils limitent le *idle listening* et le *overhearing*, et mettent les nœuds en mode sommeil durant les slots de temps réservés aux autres nœuds. De plus, ils garantissent un délai borné de bout-en-bout si un algorithme d'allocation prend en compte les caractéristiques du trafic. En revanche, l'aspect centralisé et le besoin d'une synchronisation rendent ce type de protocoles rigide et non-adapté pour les topologies dynamiques et mobiles.

Par la suite, nous décrivons brièvement les protocoles TRAMA, FLAMA et LEACH.

**TRAMA** TRAMA (*TR*affic-*A*ddaptive *M*edium *A*ccess control) [14] divise le temps en slots de temps et se base sur le trafic annoncé par les nœuds pour désigner les émetteurs et les récepteurs pour chaque slot de temps. Pour ce faire, TRAMA applique trois mécanismes :

- un protocole de voisinage appelé NP (*Neighbor Protocol*) pour échanger la liste des voisins à un saut entre les nœuds, ce qui permet d'établir une connaissance des voisins à deux sauts,
- un protocole d'échange de calendriers appelé SEP (*Schedule Exchange Protocol*) où chaque nœud annonce ce qu'il a comme trafic à envoyer en précisant les récepteurs concernés,



- un protocole d'élection adaptative AEA (*Adaptative Election Algorithm*)<sup>1</sup> qui choisit les émetteurs et les récepteurs pour un slot de temps donné en fonction des informations collectées par NP et SEP.

TRAMA découpe le temps en alternant des intervalles de temps à accès planifié et des intervalles de temps à accès aléatoire. Chaque intervalle est constitué de slots de temps. TRAMA commence par des intervalles à accès aléatoire pour permettre au réseau de s'établir. Les échanges pour la découverte de voisinage du protocole NP sont effectués durant les intervalles à accès aléatoire. Ces intervalles servent aussi pour permettre aux nouveaux nœuds de rejoindre le réseau. L'envoi de trames de données se fait durant les intervalles à accès planifié. La figure 2.1 montre comment TRAMA alterne les deux types d'intervalles de temps.

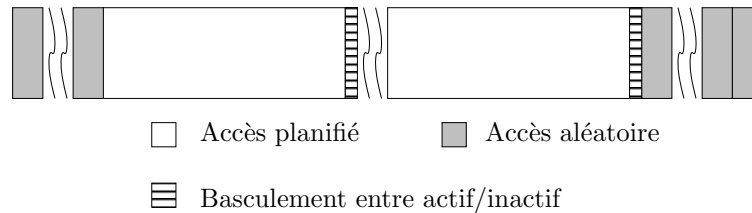


FIG. 2.1 – Découpage temporel de TRAMA pour un nœud du réseau.

Pour réaliser son découpage temporel, TRAMA a besoin que chaque nœud diffuse son calendrier de trafic et fasse une découverte de voisinage à deux sauts.

TRAMA se présente comme le premier protocole à introduire l'aspect économie d'énergie dans son mécanisme de découpage temporel en mettant en mode sommeil les nœuds non concernés par une transmission ou une réception durant un slot de temps donné. TRAMA améliore ainsi le débit utile en évitant les collisions des trames de données. En revanche, le découpage temporel séquentiel induit un délai élevé comparé aux protocoles à accès aléatoire. La découverte de voisinage se fait en accès aléatoire et génère par conséquent des collisions, de plus elle nécessite l'échange de trames de taille relativement grande par rapport à la taille des données applicatives dans les réseaux de capteurs.

**FLAMA** FLAMA (*FLow-Aware Medium Access*) [15] est proposé comme une amélioration de TRAMA. Comme TRAMA, FLAMA divise le temps selon deux modes d'activité : intervalles de temps à accès planifié et intervalles de temps à accès aléatoire. FLAMA a aussi besoin de connaître le voisinage d'un nœud à deux sauts et des informations concernant le flux de données des voisins à un saut.

En revanche, FLAMA ne diffuse pas de calendriers de trafic durant les intervalles de temps à accès planifié mais échange durant les intervalles de temps à accès aléatoire des informations par rapport aux flux de données liés à l'application. En outre, FLAMA établit une synchronisation globale du réseau basé

<sup>1</sup>Pour plus de détails concernant l'agorithme AEA, se référer à [14]

sur l'estampillage des trames et une topologie arborescente pour le relais de données.

Pour déterminer les émetteurs et les récepteurs concernés par chaque slot de temps lors de l'intervalle à accès planifié, FLAMA applique un algorithme plus simple que celui utilisé dans TRAMA <sup>2</sup>. Comme TRAMA, FLAMA assure des transmissions sans collision en ne permettant qu'à un seul nœud d'émettre dans un voisinage à deux sauts.

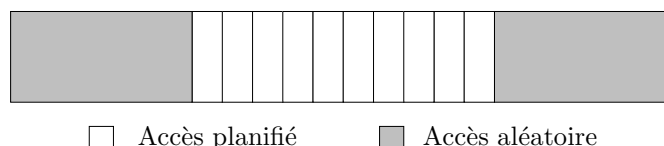


FIG. 2.2 – Découpage temporel de FLAMA.

FLAMA est plus performant que TRAMA en terme d'économie d'énergie et de perte de trames. De plus, FLAMA améliore le délai de bout en bout par rapport à TRAMA grâce à l'exploitation des routes définies à partir d'un arbre pour le cheminement multi-sauts. En revanche, FLAMA surcharge le réseau par des échanges pour connaître le flux de l'application et établir une synchronisation globale du réseau de proche en proche.

**E-MAC, L-MAC et AI-LMAC** E-MAC (*Event MAC*) [16] découpe le temps en slots, chaque slot est lui-même découpé en trois parties respectivement prévues pour accueillir les requêtes de communication, le trafic de contrôle et le trafic de données. E-MAC définit trois types de nœuds : les nœuds actifs possédant un slot, les nœuds passifs ne possédant pas de slot et ne transmettant qu'après avoir fait une requête pour utiliser le slot d'un voisin durant la partie dédiée aux requêtes de communication du slot du voisin, et les nœuds dormants qui dorment la plupart du temps et choisissent un mode passif ou actif quand ils se réveillent. Chaque nœud diffuse une information concernant les slots utilisés par ses voisins durant la partie trafic de contrôle de son slot. Les nœuds doivent tous se réveiller durant la partie trafic de contrôle des slots de leurs voisins.

L'allocation des slots commence par une station de base qui choisit un slot et annonce ce choix par une diffusion. Ensuite, ses voisins choisissent aléatoirement un slot. En cas de choix d'un même slot, les nœuds signalent ce fait durant la partie trafic de contrôle. Les slots sont réutilisés à partir de trois sauts.

L-MAC (*Lightweight-MAC*) [17] adopte le même mécanisme d'allocation de slots et de découpage temporel que E-MAC. En revanche, L-MAC force tous les nœuds à avoir au moins un slot. Ainsi, la partie requête de communication du slot n'est plus présente dans L-MAC et un slot est fractionné en 2 parties seulement.

AI-LMAC (*Adaptive, Information-centric and Lightweight MAC*) [18] est une amélioration de L-MAC

---

<sup>2</sup>Pour plus de détails concernant cet algorithme, se référer à [15]

qui prend en compte les conditions du trafic applicatif et offre la possibilité aux nœuds d'avoir plusieurs slots. AI-LMAC regroupe les nœuds avec des relations père-fils. Le père surveille les conditions de trafic de ses fils et demande à ses fils de s'allouer plus de slots ou de se désallouer des slots selon leur charge.

La figure 2.3 montre le découpage temporel en slots des trois protocoles.

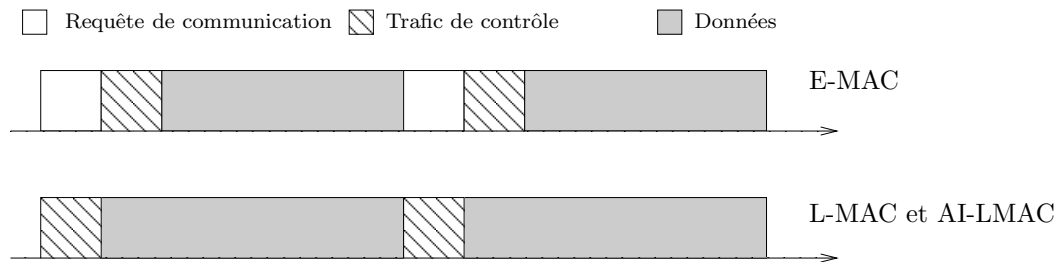


FIG. 2.3 – Découpage temporel de E-MAC, L-MAC et AI-MAC.

Le mécanisme d'allocation de slots peut prendre beaucoup de temps à cause des conflits d'allocation pour servir tous les nœuds du réseau. Dans AI-LMAC, l'*overhead* généré pour maintenir une mise à jour des informations concernant les conditions de trafic est critique en terme d'utilisation mémoire, d'échange de trames de contrôle, de consommation énergétique et de temps de calcul.

### 1.2.2 Protocoles basés sur la contention

Dans ce type de protocoles MAC, les nœuds accèdent au médium durant le même intervalle de temps en utilisant un algorithme de la famille CSMA/CA, chacune de ses variantes essayant d'éviter les collisions.

Le point fort de ce type de protocoles est sans doute l'extensibilité et le passage à l'échelle. En revanche, ces protocoles n'offrent pas de délai borné du fait qu'ils ne garantissent pas l'accès au médium dès que la charge du réseau augmente.

Nous allons commencer cette partie en décrivant brièvement l'algorithme de CSMA/CA de la norme IEEE 802.11 en mode DCF, sur lequel est basée la plupart des protocoles qui exploitent de différentes façons une période de contention, ce que nous allons décrire par la suite.

**CSMA/CA de la norme IEEE 802.11** CSMA/CA est une méthode d'accès de la même famille que CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) [19] puisqu'elle impose à un émetteur de s'assurer que le canal est libre avant d'émettre. Dans CSMA/CA, les collisions ne peuvent pas être détectées comme dans le CSMA/CD, un nœud essaie d'éviter les collisions (sans vraiment les éviter à 100 %). Ceci à cause de l'effet d'aveuglement du médium sans fil (*Near Far Effect*) qui empêche une entité de recevoir quand elle est en train d'émettre.

Deux modes de fonctionnement existent dans la norme IEEE 802.11 [20] [21] : le mode infrastructure

où les entités communiquent via un nœud central appelé point d'accès, et le mode ad hoc où les nœuds peuvent communiquer entre eux, quand la topologie le permet, sans passer par un point d'accès. Il existe aussi deux techniques d'accès au canal : le mode DCF (*Distributed Coordination Function*) où le CSMA/CA est employé, utilisé dans les architectures de type ad hoc et infrastructure, et le mode PCF (*Point Coordination Function*) utilisé uniquement dans les architectures de type infrastructure où l'accès au canal est organisé selon une structure temporelle appelée supertrame et délimitée par des *beacons*. Nous nous intéressons uniquement au mode DCF et par la suite, nous allons décrire brièvement comment la scrutation du canal se fait avant d'accéder au canal et expliquer le problème du terminal caché et le terminal exposé dans ce mode.

**Clear Channel Assessment** Un nœud qui souhaite envoyer une trame doit s'assurer que le canal est logiquement libre dans son environnement géographique, pour cela il effectue un CCA. On appelle CCA (*Clear Channel Assessment*) le test effectué pour détecter si le canal est libre ou occupé. Le canal est considéré occupé quand la puissance détectée est supérieure à un certain seuil. Ce seuil est appelé le seuil de détection de porteuse ou seuil d'interférence. Il existe un autre seuil, le seuil de réception, qui définit la portée d'une station. Celui-ci correspond au niveau de puissance au dessous duquel la trame n'est plus décodable par le récepteur. Ce seuil dépend essentiellement du débit de transmission. En diminuant le débit, le seuil de réception diminue et le signal est décodable de plus loin. Dans le cas de 802.11, avec un débit de 11 *Mbits/s* ce seuil est égal à  $-82 \text{ dBm}$ <sup>3</sup>. En revanche, le seuil d'interférence est indépendant du débit et est égal à  $-95 \text{ dBm}$ .

La zone comprise entre ces deux seuils est appelée zone d'interférence<sup>4</sup>. La figure 2.4 montre les deux seuils et différents cas de réception selon la position du récepteur par rapport à l'émetteur.

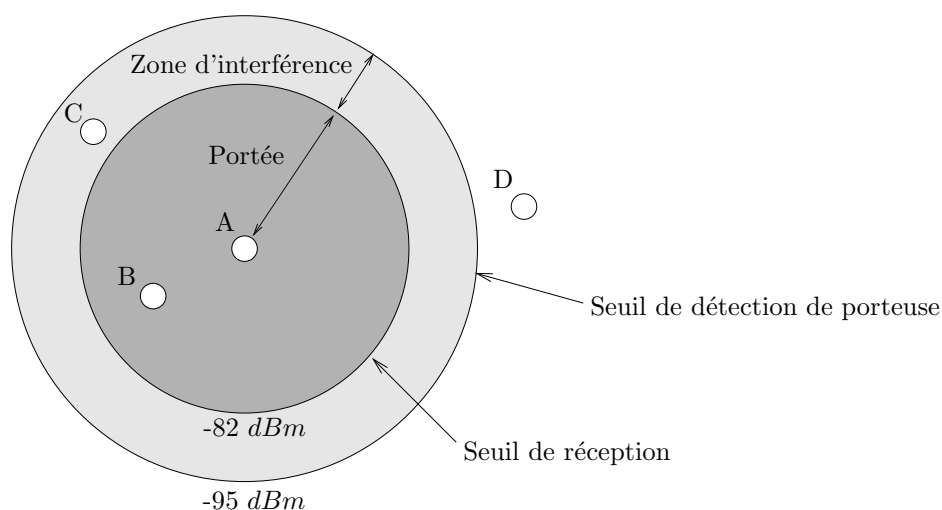


FIG. 2.4 – Portée et zone d'interférence.

<sup>3</sup>Dans le cas de ZigBee avec un débit de 250 *Kbits/s* ce seuil est égale à  $-92 \text{ dBm}$

<sup>4</sup>Nous avons effectué des mesures que nous présentons dans l'annexe A pour identifier cette zone.

Dans la figure 2.4, le nœud B reçoit très bien les signaux envoyés par le nœud A. Le nœud C ne peut pas décoder les signaux de A. De plus, le nœud C est gêné par les signaux du nœud A qui l'empêchent d'accéder au médium. Le nœud D ne détecte aucun signal significatif indiquant que A est en train d'émettre.

**Accès au médium** Si le canal est détecté libre, suite à un CCA, pour une durée DIFS (*DCF InterFrame Spacing*) le nœud accède au canal. Sinon, le nœud tire un nombre aléatoire de slots de temps dit *backoff*. Ensuite, le nœud décrémente ce *backoff* pour chaque slot de temps pendant lequel le nœud détecte que le canal est libre et émet sa trame quand le compteur de *backoff* expire. Le nœud suspend la décrémentation quand il détecte une activité avant que le compteur de *backoff* expire et reprend la décrémentation quand le canal devient libre pour une durée DIFS.

Quand le nœud récepteur reçoit correctement la trame il répond par une trame d'acquittement après avoir respecté une durée SIFS (*Short InterFrame Spacing*) .

**Terminal caché et terminal exposé** le CSMA/CA est efficace pour éviter les collisions à un saut, mais il ne résout pas le problème à plusieurs sauts. Deux cas de figure se posent pour lesquels le CSMA/CA n'est pas fonctionnel et cause une forte dégradation des performances : le terminal caché et le terminal exposé. La figure 2.5 montre un scénario du cas du terminal caché. Le nœud C est hors la zone d'interférence de A et ne peut pas détecter l'activité de A. Quand C effectue son CCA pour envoyer un message à B, il trouve le médium libre alors que A est en train de communiquer avec B. Le nœud A est alors un terminal caché par rapport à C. La figure 2.6 montre une situation de terminal exposé. Le nœud B veut communiquer avec C mais ne le fait pas car il détecte que le médium est occupé à cause de la communication de A vers D. B n'a pas pu effectuer son envoi alors que C était prêt à recevoir sans risque de collision. Dans ce cas, B est le terminal exposé aux transmissions de A.

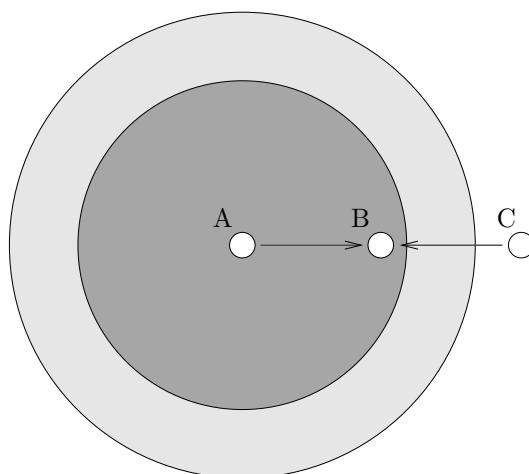


FIG. 2.5 – A et C envoient simultanément deux messages à B et génèrent un risque de collision en B.

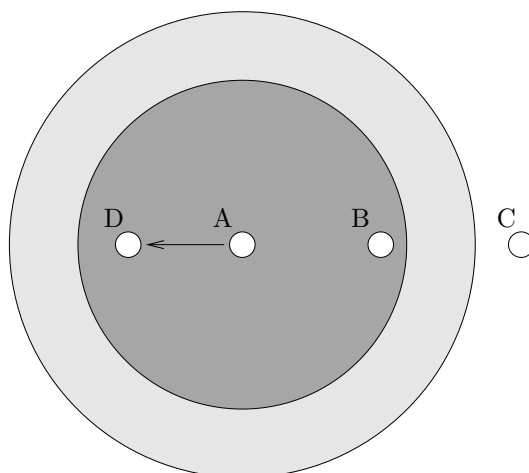


FIG. 2.6 – A communique avec D. B veut communiquer avec C mais ne le fait pas parce que A occupe le médium, alors que si B communique avec C aucune collision n'aura lieu.

Pour remédier au problème du terminal caché, la norme 802.11 utilise le mécanisme du RTS/CTS (*Request To Send/Clear To Send*). Dans le cas de la figure 2.5, quand C trouve le médium libre, avant d'envoyer les trames de données, il envoie une trame RTS à B et attend la réception de CTS provenant de B. De cette façon si B ne répond pas par un CTS, C n'envoie pas ses trames de données. L'écoute des messages RTS/CTS qui servent à réserver localement le canal en indiquant la longueur de la trame à venir permet de réduire les risques de collision.

**S-MAC, T-MAC et D-MAC** S-MAC [22] (*Sensor-MAC*) est conçu pour assurer une méthode d'accès économe en énergie pour les réseaux de capteurs sans fil. Pour ce faire, les nœuds se mettent en mode sommeil pendant une certaine durée et se réveillent pour écouter le médium pendant une autre durée.

Les nœuds échangent leur calendrier de périodes d'écoute en le diffusant à leurs voisins à un saut. Ainsi, chaque nœud connaît le calendrier de ses voisins et sait quand il faut se réveiller pour communiquer avec un nœud à sa portée. Plusieurs nœuds peuvent avoir le même intervalle de temps comme période d'écoute. Les nœuds accèdent au médium en utilisant le CSMA/CA de IEEE 802.11 avec le mécanisme RTS/CTS. En outre, un champ supplémentaire est ajouté à tous les messages (y compris les messages RTS/CTS et les acquittements) indiquant la durée de l'échange, ce qui permet aux nœuds non concernés de dormir pendant cette durée.

Pour maintenir une synchronisation des horloges, les nœuds émetteurs envoient des messages de synchronisation SYNC au début de la période d'écoute de leurs voisins.

Dans [23], les auteurs améliorent le fonctionnement de S-MAC en minimisant le délai de bout-en-bout. Pour ce faire, ils obligent les nœuds, après avoir reçu un RTS ou un CTS qui ne les concerne pas, à se réveiller pour une courte durée après la fin de la transmission pour vérifier s'ils sont la prochaine destination et recevoir la trame si c'est le cas.

La figure 2.7 montre le séquençement des périodes d'écoute et de sommeil des nœuds avec le découpage en deux parties de la période d'écoute. Les émetteurs B et C, qui souhaitent communiquer avec le récepteur A, connaissent la période d'écoute de A grâce aux messages SYNC envoyés par A.

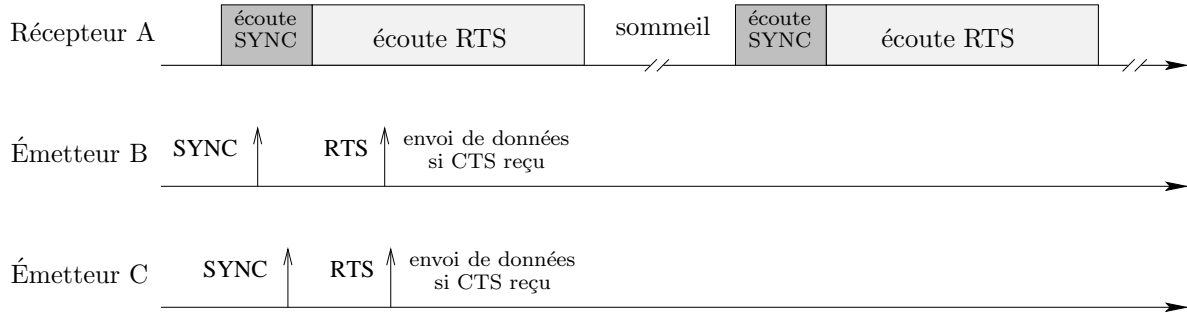


FIG. 2.7 – Séquençement des périodes d'écoute et de sommeil dans S-MAC.

S-MAC apporte une amélioration par rapport au CSMA/CA de la norme 802.11 en terme d'économie d'énergie. Cependant, les messages de synchronisation et les messages RTS/CTS génèrent une surcharge du réseau d'autant plus grande que la longueur des messages de données échangés dans les réseaux de capteurs est relativement courte.

T-MAC [24] (*Timeout-MAC*) propose de mettre un nœud en mode sommeil après un temps TA durant lequel le nœud n'a reçu aucun message. Ainsi, T-MAC réduit l'*idle listening* par rapport à S-MAC. La figure 2.8 montre le mode de fonctionnement de T-MAC par rapport à celui de S-MAC pour un même nœud.

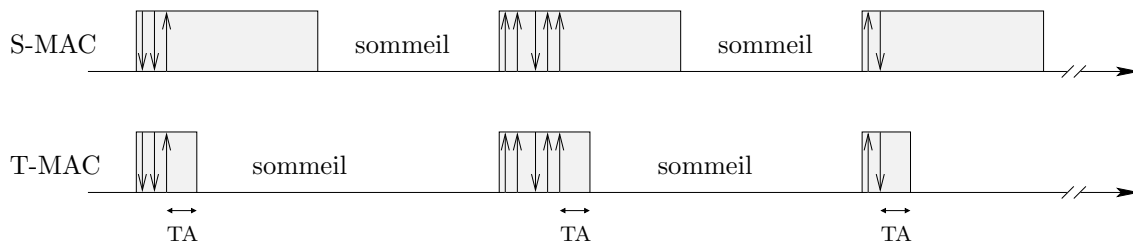


FIG. 2.8 – Séquençement des périodes d'écoute et de sommeil de T-MAC et de S-MAC.

Avec cette approche, T-MAC fait dormir un nœud sans s'être assuré que ses voisins n'ont plus de données à lui envoyer. En effet, les données à envoyer du voisin ont pu être retardées à cause d'un échec d'accès au canal. Ce problème est appelé le sommeil prématuré dans [24]. T-MAC propose d'envoyer un FRTS (*Future Request To Send*) après la réception d'un CTS. Ce FRTS informe la destination qu'elle ne doit pas dormir après TA. Pour éviter une collision entre les données et le FRTS, le nœud qui devait envoyer après le CTS reporte sa transmission pour la durée d'envoi d'un FRTS. Une autre solution est de prendre l'initiative en transmettant un RTS à la réception d'un RTS. Au lieu de répondre par un CTS, le nœud peut décider d'envoyer un RTS pour éviter que le destinataire des messages de ce nœud

dorme au bout de TA (pour plus de détail sur ces mécanismes se référer à [24]).

Dans [24], il est montré que T-MAC réduit la consommation énergétique jusqu'à 96% sous faible charge par rapport à CSMA/CA de 802.11, et que T-MAC consomme autant que S-MAC sous forte charge. En revanche, T-MAC génère plus de trafic de contrôle que S-MAC.

Dans S-MAC et T-MAC, les nœuds choisissent leur période de réveil soit aléatoirement, soit de manière à ce qu'elle coïncide avec celle d'un voisin. Cela induit des retards sur le délai de bout-en-bout.

D-MAC [25] (*Data gathering MAC*) propose un séquençement des périodes d'activité qui favorise la collecte d'informations dans une topologie arborescente. Les nœuds de même niveau se réveillent en même temps. Les fils d'un nœud accèdent au médium en même temps en utilisant des délais (appelés *backoff*) aléatoires pour éviter les collisions systématiques. De ce fait, D-MAC sous-estime la probabilité de collisions en supposant que les nœuds fils peuvent se détecter. D-MAC suppose aussi que la durée nécessaire pour envoyer le trafic cumulé dans les nœuds les plus proches de la racine reste inférieure à la durée du cycle d'activité.

**LPL : B-MAC, WiseMAC, B-MAC+, X-MAC et DW-LPL** Le LPL (*Low Power Listening*) est l'une des premières approches pour réduire l'*idle listening* en introduisant une période d'inactivité au niveau de la couche physique. L'idée est de pénaliser les émetteurs en envoyant un préambule long pour économiser l'énergie des récepteurs. Les récepteurs activent leur module radio périodiquement pour détecter la présence d'un préambule. La durée de transmission du préambule doit être de la même longueur que l'intervalle entre deux réveils d'un récepteur. La figure 2.9 montre comment le nœud B s'active de temps en temps et détecte un préambule envoyé par A. LPL est un mécanisme qui peut être ajouté à la plupart des protocoles MAC.

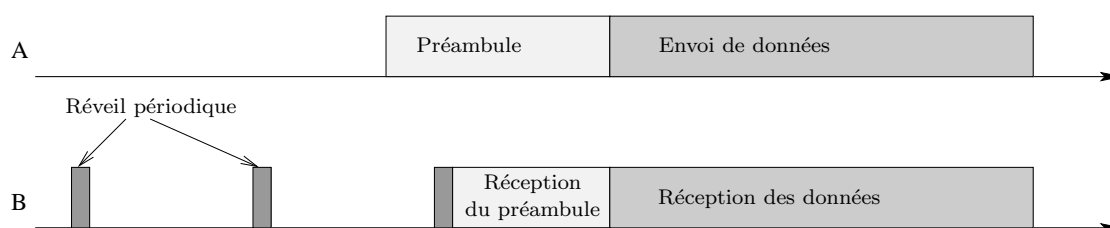


FIG. 2.9 – LPL : L'émetteur utilise un long préambule pour permettre au récepteur d'activer son module radio seulement de temps en temps.

L'un des premiers protocoles MAC pour les réseaux de capteurs sans fil basés sur cette technique est B-MAC (*Berkeley-MAC*) [26]. Dans [27], des résultats de simulation ont permis de constater que la technique du LPL réduit l'*idle listening* et améliore les performances en terme d'économie d'énergie de S-MAC et T-MAC.

WiseMAC [28] a été proposé pour réduire la longueur du préambule en fonction des périodes de réveil



des récepteurs voisins. De cette façon, WiseMAC réduit l'*overhead* du long préambule et l'*overhearing* engendré par ce préambule, étant donné que tous les nœuds doivent rester éveillés à la réception d'un préambule pour attendre la trame de données, afin de savoir s'ils sont destinataires du message ou non.

B-MAC+ [29] améliore B-MAC en remplaçant le long préambule par plusieurs petits messages appelés des paquets de *countdown* contenant chacun l'identifiant du destinataire et le temps restant pour commencer l'envoi des données. Cela permet aux nœuds non concernés de se remettre en mode sommeil en détectant un petit préambule au lieu de rester éveillé pour la durée de la transmission du long préambule traditionnel.

X-MAC [30] décompose lui aussi le long préambule en plusieurs préambules de petite taille incluant l'identifiant du destinataire du message. Contrairement B-MAC+, le destinataire prévient l'émetteur de son écoute avec un acquittement envoyé entre deux préambules consécutifs. L'émetteur peut alors commencer la transmission des données. Cela réduit l'*overhead* au niveau de l'émetteur qui arrête la transmission du préambule dès la réception de l'acquittement.

DW-LPL (*Dual Wake-up LPL*) [31] limite le problème de l'*overhearing* en employant la technique du LPL pour les messages diffusés uniquement. L'envoi de messages *unicast* utilise une technique de signalement par des trames spécifiques envoyés par les récepteurs pour informer leur entourage qu'ils sont prêts à recevoir des trames en *unicast*. Les instants et la périodicité d'envoi des messages de signalement sont indépendants d'un nœud à l'autre. Ceci, en contre partie, génère des collisions dues au problème du terminal caché.

### 1.2.3 Protocoles hybrides

Une troisième famille de protocoles propose de combiner les deux méthodes : TDMA et CSMA/CA. Ainsi, ces protocoles essaient d'avoir les avantages des deux méthodes en alternant les deux dans le temps ou en les combinant d'une manière intelligente. Dans la suite, nous allons décrire trois protocoles hybrides : Z-MAC, G-MAC et Funneling-MAC.

**Z-MAC** Z-MAC [32] (*Zebra-MAC*) est un protocole hybride qui alterne des périodes de TDMA et CSMA/CA selon le nombre d'entités en concurrence en un instant donné. Z-MAC utilise un découpage temporel basé sur TDMA et gère les accès durant les slots avec le CSMA/CA de IEEE 802.11.

Une fois le réseau déployé, Z-MAC commence par une phase de découverte du voisinage à deux sauts suivie par une assignation de slots aux nœuds en utilisant DRAND (*Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks*) [33]. DRAND est un protocole distribué qui assure qu'un slot de temps n'est pas assigné à deux nœuds situés à moins de trois sauts l'un de l'autre. La synchronisation nécessaire est obtenue à l'aide de deux protocoles utilisés de façon complémentaire.

Pour accéder au médium, si le nœud est le propriétaire du slot courant, il attend un temps aléatoire plus petit qu'une valeur  $T_o$  puis effectue un CCA. Si le canal est libre, il émet. Sinon, il attend que le canal devienne libre et il recommence la même démarche. Si le slot actuel appartient à un voisin à deux sauts et si le nœud a reçu une indication de forte contention d'un de ses voisins à deux sauts, le nœud n'a pas le droit d'utiliser ce slot. Sinon, il attend un temps aléatoire compris entre  $T_o$  et  $T_{no}$  avant d'effectuer un CCA.

La figure 2.10 montre le découpage temporel pour une topologie simpliste représentée par quatre nœuds : A, B, C et D. Notons que D est à 3 sauts de A. Ceci permet à A et D de partager le même intervalle de temps.

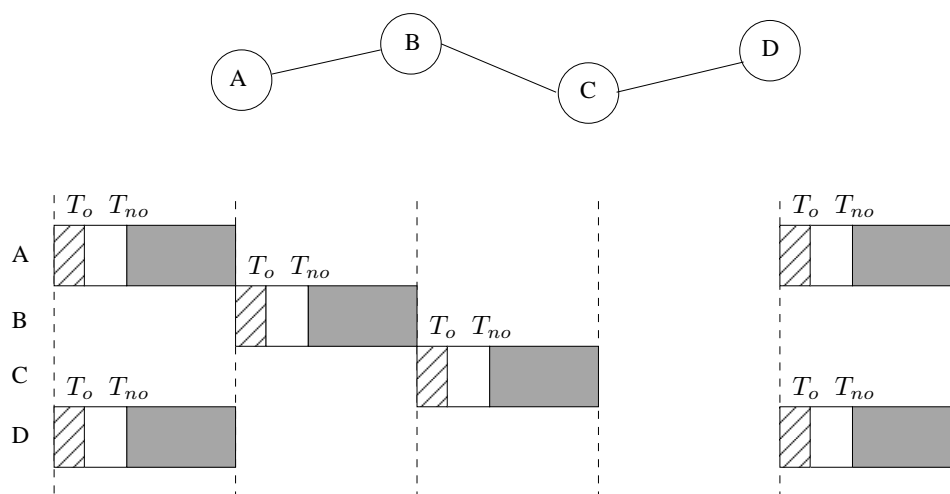


FIG. 2.10 – Découpage temporel de Z-MAC. Notons l'existence d'un temps pendant lequel le médium n'est pas utilisé.

Z-MAC utilise la technique LPL. Comme les nœuds peuvent émettre durant tous les intervalles, les nœuds écoutent le médium périodiquement pour vérifier s'il y a des émissions qui les concernent.

Z-MAC est plus performant que B-MAC sous forte et moyenne charge, et moins performant sous faible charge, surtout en terme d'économie d'énergie.

Z-MAC est fait pour optimiser l'utilisation du canal en se comportant comme CSMA/CA sous faible charge et comme TDMA sous forte charge. Le CSMA/CA utilisé par Z-MAC n'est pas économe en énergie, car chaque nœud écoute le médium tant qu'il est détecté occupé. Z-MAC n'emploie pas un mécanisme de report afin de ne pas déborder sur le slot suivant et ne garantit pas qu'un seul nœud soit propriétaire d'un slot. Cela est dû aux imprécisions de synchronisation et aux erreurs potentielles d'assignation de slots. Une réutilisation de slots à partir de deux sauts génère des collisions à cause des acquittements de niveau MAC.

**G-MAC** G-MAC (*Gateway MAC*) [34] organise les échanges à l'intérieur d'un *cluster*. Dans ce domaine, un *cluster* est un ensemble de nœuds géré par une station appelée passerelle. G-MAC découpe le temps en deux périodes : une période de collecte où les échanges se font en CSMA/CA, et une période de distribution où les échanges se font en TDMA.

Durant la période de collecte, les nœuds envoient à leur passerelle deux types de trafic : les requêtes FRTS (*Future Request To Send*) pour réserver un slot de temps dans la période de distribution pour échanger avec d'autres nœuds appartenant au même *cluster*, et le trafic inter-*cluster* destiné aux nœuds appartenant à des *clusters* différents. Durant un slot, les échanges se font selon le séquençement RTS-CTS-données-acquittement. La période de distribution commence par une diffusion d'un message GTIM (*Gateway Traffic Indication Message*) qui contient les indications temporelles des deux périodes suivantes ainsi que le séquençement des échanges entre les nœuds du *cluster* qui ont réussi à envoyer une requête FRTS.

Pendant la période de collecte et pendant les slots non réservés de la période de distribution la passerelle échange les données inter-*cluster* avec d'autres passerelles une fois la collecte terminée.

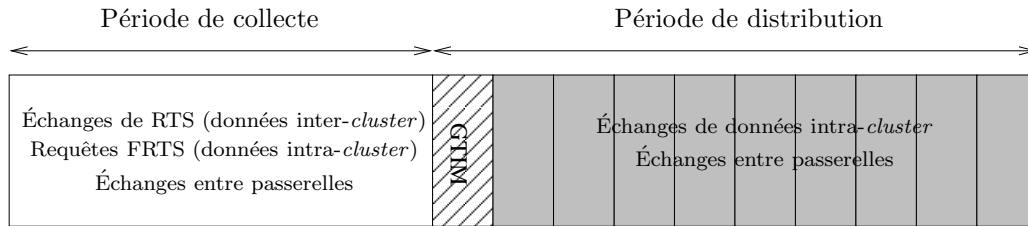


FIG. 2.11 – Découpage temporel de G-MAC.

Dans [35], les auteurs comparent G-MAC à 802.11, B-MAC, S-MAC et T-MAC. Ils montrent que G-MAC double la durée de vie du réseau dans le pire des *scenarii*. En revanche, dans cette évaluation la taille du réseau est limitée à la taille d'un *cluster* et donc à la portée de la passerelle, mais suppose en même temps l'existence de plusieurs *clusters* et le fait que les passerelles échangent entre elles les messages inter-*cluster*. Néanmoins, si les *clusters* sont à portée les uns des autres cela génère des conflits de réservation des slots TDMA dans la période de distribution.

**Funneling-MAC** Funneling-MAC [36] est un protocole MAC qui prend en compte le goulot d'étranglement dont souffrent la plupart des applications des réseaux de capteurs. Ce phénomène survient quand une station du réseau joue le rôle d'un puits de données vers lequel un ensemble de capteurs dirige son trafic. Cela est représenté sur la figure 2.12 sur laquelle est identifiée une zone à forte charge.

Funneling-MAC adopte une méthode d'accès en CSMA/CA dans l'ensemble du réseau durant un intervalle de temps suivi par un intervalle de temps durant lequel une méthode d'accès en TDMA est utilisée pour la zone à forte charge uniquement pour offrir plus de temps d'accès aux nœuds à proximité

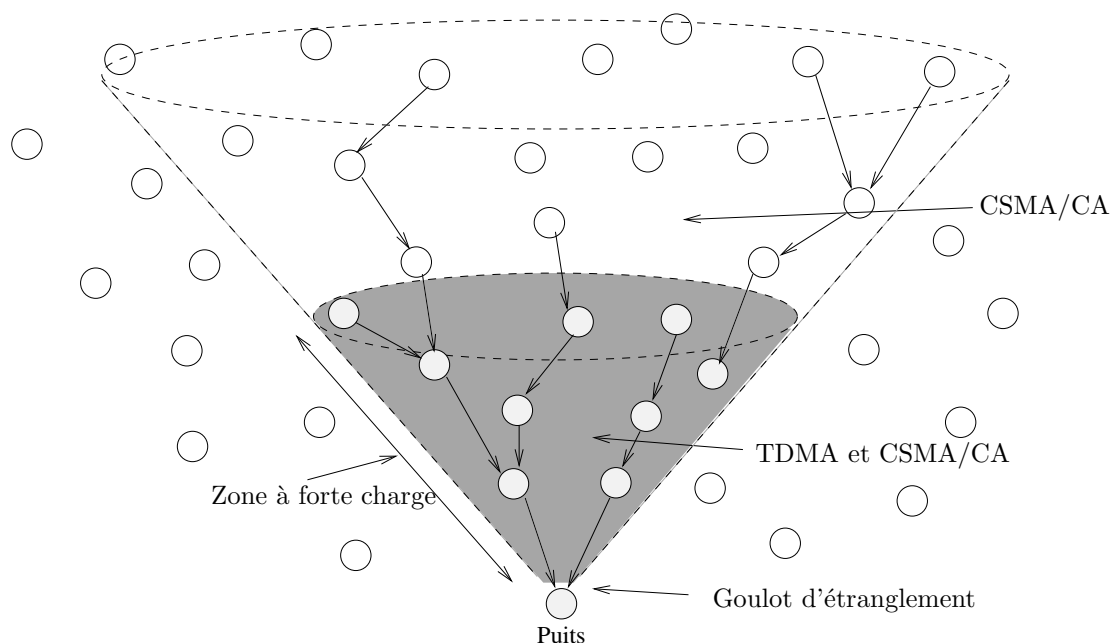


FIG. 2.12 – Gestion du goulot d'étranglement dans Funneling-MAC.

du puits. Ces deux intervalles de temps constituent une supertrame. Ce découpage est représenté sur la figure 2.13.

La zone à forte charge est dimensionnée par une diffusion d'un *beacon* par le nœud puits. Les nœuds qui ne reçoivent pas ce *beacon* appliquent le CSMA/CA.

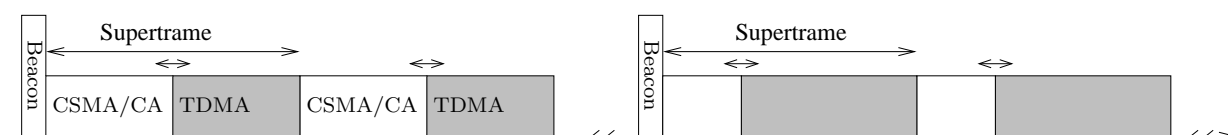


FIG. 2.13 – Découpage temporel dans Funneling-MAC.

Le puits se base sur le chemin parcouru par les trames qu'il reçoit pour définir le séquençement et le dimensionnement des slots TDMA alloués aux nœuds de la zone à forte charge. Seuls les nœuds appartenant à la zone à forte charge mettent à jour le chemin parcouru par une trame.

Pour éviter que les nœuds qui se trouvent au delà de la zone à forte charge interfèrent avec le découpage temporel du TDMA et le *beacon* transmit par le nœud puits, les nœuds de la zone à forte charge génèrent périodiquement une trame contenant les informations concernant la durée de la période CSMA/CA, la durée de la période TDMA et le nombre de supertrames jusqu'au prochain *beacon*.

Funneling-MAC s'avère plus performant que B-MAC et Z-MAC en terme de débit, d'économie d'énergie et de nombre de paquets de contrôle. Funneling-MAC considère que le puits possède plus de capacité de calcul et de ressources énergétiques que les autres nœuds du réseau et qu'il est donc capable d'exécuter les différents algorithmes du protocole.

#### 1.2.4 Récapitulatif des caractéristiques des protocoles présentés

Le tableau 2.2 récapitule, d'une manière globale, l'apport de chacun des protocoles présentés concernant les 2 besoins essentiels des applications industrielles : l'économie d'énergie et la qualité de service.

Nom du protocole	Économie d'énergie	Qualité de service
<b>Protocoles basés sur TDMA</b>		
TRAMA FLAMA E-MAC L-MAC AI-LMAC	Un nœud économise de l'énergie durant les slots dans lesquels il n'est ni émetteur ni récepteur	Accès garanti suite à une réservation en accès non-garanti
<b>Protocoles basés sur CSMA/CA</b>		
S-MAC	Tous les nœuds peuvent économiser de l'énergie	Aucune
T-MAC	Améliore l'économie d'énergie de S-MAC [37]	Aucune
D-MAC	Économise plus d'énergie que S-MAC [25]	Diminue le délai de bout-en-bout par rapport à S-MAC
<b>Protocoles basés sur LPL</b>		
B-MAC WiseMAC B-MAC+ X-MAC DW-LPL	Améliorent l'économie d'énergie des protocoles CSMA/CA [30] [31]	Aucune
<b>Protocoles Hybrides</b>		
Z-MAC	Plus économe sous forte charge que B-MAC [32]	Réduit le déterminisme du TDMA de base pour améliorer l'utilisation du canal
G-MAC	Limitée aux nœuds à portée d'un nœud central	Limitée aux nœuds à portée d'un nœud central
Funneling-MAC	Plus économe sous forte charge que Z-MAC [36]	Limitée aux nœuds proches d'un nœud central

TAB. 2.2 – Récapitulatif des protocoles étudiés en terme d'économie d'énergie et de qualité de service.

Nous pouvons constater qu'aucun protocole ne garantit l'économie d'énergie et la qualité de service en même temps. Il y a toujours un compromis à faire entre ces deux aspects. Un autre compromis est aussi la complexité des algorithmes utilisés pour assurer le bon fonctionnement de ces protocoles, notamment les protocoles ayant des périodes basées sur TDMA, plus les algorithmes sont simples, moins intéressant est leur rendu en terme de performances de réseaux.

## 2 La norme IEEE 802.15.4/ZigBee

La bibliographie indique différentes pistes et mécanismes intéressants pour les spécifications d'une couche MAC adaptée aux réseaux de capteurs sans fil économes en énergie et capable d'offrir une certaine qualité de service. Néanmoins, la pérennité d'une solution est très souvent consistante à partir d'une norme. La norme IEEE 802.15.4 [38] définit la couche MAC et la couche physique pour les réseaux personnels sans fil à bas débit, aussi appelés *Low Rate Wireless Personal Area Networks* en anglais (LR-WPAN). Cette norme convient très bien aux besoins des réseaux de capteurs sans fil en terme de bas débit, faible consommation énergétique et faible coût des entités du réseau.

Deux types d'entités sont présentes dans ce réseau : les FFD (*Full Function Device*) et les RFD (*Reduced Function Device*). Un FFD peut fonctionner selon trois profils différents dans le réseau : celui d'un coordinateur du PAN (*Personal Area Network*), d'un coordinateur ou d'une feuille. Un RFD en revanche ne peut être qu'une feuille, et donc ne contient pas toutes les fonctionnalités prévues pour la couche MAC 802.15.4. Un coordinateur peut communiquer avec toute autre entité du réseau alors qu'un RFD ne peut communiquer qu'avec un FFD. Les RFD sont prévus pour des rôles applicatifs simples comme un capteur infrarouge passif chargé de faire une détection de présence ou un interrupteur commandant une source lumineuse. Un RFD est associé à un seul FFD à un instant donné.

La norme IEEE 802.15.4 supporte deux types de topologies présentés sur la figure 2.14 : les topologies en étoile et les topologies pair-à-pair. En topologie en étoile, les stations communiquent uniquement avec un nœud central, qui est le coordinateur du PAN et la seule entité à pouvoir accepter de nouvelles associations dans le réseau. La taille du réseau est limitée à la portée du coordinateur du PAN. Ce type de topologies est typiquement utilisé pour les applications domotiques, gestion des périphériques, etc.

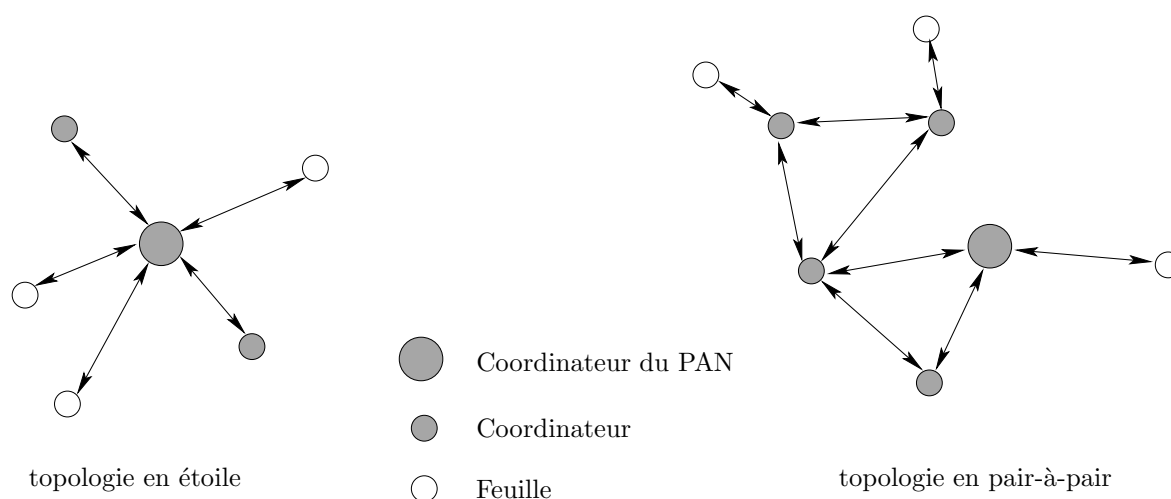


FIG. 2.14 – Exemples de topologies en étoile et en pair-à-pair.

Dans une topologie pair-à-pair, nous avons toujours un coordinateur de PAN, mais tous les RFD

sont capables de communiquer les uns avec les autres, à condition qu'ils soient à portée. Contrairement à la topologie en étoile, les coordinateurs fils du coordinateur du PAN permettent à d'autres entités de s'associer au réseau à travers eux. Cela permet la formation d'un réseau plus complexe et plus large comme les réseaux maillés. Parmi les applications qui profitent d'une topologie pair-à-pair, nous trouvons les applications de surveillance et maintenance industrielles et les applications de traçabilité et de localisation.

La norme IEEE 802.15.4 est fortement liée au protocole ZigBee [39]. En effet, la ZigBee Alliance [40], qui est une organisation internationale, a normalisé une pile réseau complète. Cette dernière s'appuie sur les résultats d'un groupe de travail IEEE, en utilisant la norme IEEE 802.15.4 pour les couches physique et MAC et en spécifiant les couches réseau et application. La figure 2.15 représente d'une manière simplifiée la pile IEEE 802.15.4/ZigBee. Chaque couche dispose de deux interfaces pour communiquer avec la couche supérieure : une interface pour les échanges des primitives de données (*Data Entity*) et une interface pour les échanges des primitives de gestion (*Management Entity*).

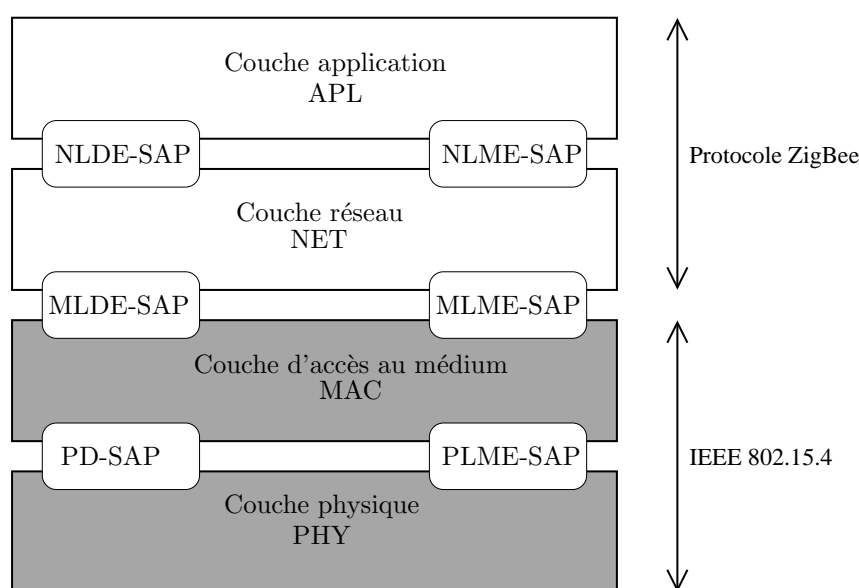


FIG. 2.15 – La pile IEEE 802.15.4/ZigBee.

La listes des interfaces est la suivante :

- NLDE pour *Network Layer Data Entity-Service Access Point* pour les échanges de primitives de données entre la couche réseau et la couche application,
- NLME pour *Network Layer Management Entity* pour les échanges de primitives de contrôle entre la couche réseau et la couche application,
- MLDE pour *MAC Layer Data Entity* pour les échanges de primitives de données entre la couche MAC et la couche réseau,
- MLME pour *MAC Layer Management Entity* pour les échanges de primitives de contrôle entre la

couche MAC et la couche réseau,

- PLME pour *PHY Layer Management Entity* pour les échanges de primitives de contrôle entre la couche physique et la couche MAC,
- PD pour *PHY layer Data* pour les échanges de primitives de données entre la couche physique et la couche MAC.

Dans la suite de cette partie, nous décrivons les fonctionnalités principales de chacune de ces quatre couches.

## 2.1 Couche physique IEEE 802.15.4

La couche physique IEEE 802.15.4 est généralement prise en charge par le module radio. Elle offre quatre débits différents. Le tableau 2.3 résume les débits proposés selon la fréquence et la modulation<sup>5</sup>.

Fréquence (MHz)	Modulation	Débit (kb/s)
868/868.6	BPSK	20
868/868.6	ASK	250
868/868.6	O-QPSK	100
902/928	BPSK	40
902/928	ASK	250
902/928	O-QPSK	250
2400/2483.5	O-QPSK	250

TAB. 2.3 – Le débit en fonction de la fréquence et de la modulation.

Dans la bande de fréquences des 2.4 GHz, le débit est donc de 250 Kb/s, il lui est associé un seuil de réception qui va jusqu'à  $-92$  dBm.<sup>6</sup>

Avec les trois plages de fréquences, la couche physique offre 27 canaux de transmission différents dont 16 sur la plage de fréquences de 2400/2483.5 MHz séparés de 5 MHz, 10 sur la plage de fréquences de 902/928 séparés de 2 MHz et 1 canal sur la plage de fréquences de 868/868.6 MHz. L'usage de ces canaux dépend de la législation des pays dans lesquels des solutions conformes à ce standard sont utilisées.

Parmi les fonctionnalités de contrôle de cette couche, nous pouvons disposer de celles qui permettent de :

- activer et désactiver le module radio,
- remonter l'état d'un lien à la couche supérieure,
- tester l'occupation du canal en faisant un CCA,

---

<sup>5</sup>Ces modulations sont définies dans [41].

<sup>6</sup>À noter que le débit utilisé pour l'envoi des trames de données par la couche physique de la norme IEEE 802.11 est de 11 Mb/s, ce qui donne un seuil de réception de  $-82$  dBm. Le seuil de détection de porteuse est de  $-95$  dBm.



- choisir le canal de transmission.

### 2.1.1 Activation et désactivation du module radio

Le module radio possède trois états de fonctionnement : un état de transmission, un état de réception et un état de sommeil. Le temps de changement d'état entre transmission et réception ne doit pas dépasser les 192  $\mu s$ . Cet état est piloté par la couche MAC. Il est essentiel pour économiser de l'énergie de mettre le module en mode sommeil.

### 2.1.2 Indication de la qualité du lien

Le LQI (*Link Quality Indication*) caractérise la qualité d'un lien à un instant donné suite à une réception d'une trame. Ce paramètre est essentiel pour les protocoles des couches réseau et application. Par exemple, un protocole de routage peut exploiter cette indication afin d'identifier les meilleurs liens à utiliser dans son choix de routes.

### 2.1.3 Test d'occupation du médium ou CCA (*Clear Channel Assessment*)

Le CCA permet de savoir l'état du canal radio. Il est essentiel pour le fonctionnement de l'algorithme de CSMA/CA de la couche MAC. La couche physique est capable d'effectuer trois modes de CCA différents :

- La détection d'un signal avec une puissance reçue supérieure à un certain seuil.
- La détection d'un signal conforme à la modulation de la couche physique.
- La détection d'un signal qui répond aux deux conditions.

À noter que le seuil de détection d'activité est typiquement -95 *dBm*.

### 2.1.4 Sélection du canal

Comme la couche physique offre plusieurs canaux de transmission, il est nécessaire de sélectionner un canal précis, ceci à la demande des couches supérieures. Le changement de canal est aussi fait implicitement par la couche physique suite à une demande de scrutation sur l'ensemble des plages de fréquences chacune constituant un canal de transmission. Cette action est appelée un *scan*.

## 2.2 Couche MAC IEEE 802.15.4

La couche MAC 802.15.4 supporte deux modes de fonctionnement selon les besoins applicatifs : le mode suivi de *beacon* et le mode non suivi de *beacon*. En mode suivi de *beacon*, le coordinateur envoie périodiquement un *beacon* pour synchroniser l'activité des entités qui lui sont attachées selon une

structure de supertrame donnée sur la figure 2.16. En mode non suivi de *beacon*, les *beacons* ne sont utilisés que pour la découverte du réseau.

Par la suite, nous allons décrire les fonctionnalités de gestion essentielles de la couche MAC 802.15.4 :

- l'accès au médium,
- les *scans*, la création du réseau et l'association,
- la synchronisation avec un coordinateur,
- les échanges de trames.

### 2.2.1 Accès au médium

Un coordinateur limite l'accès au médium en utilisant la diffusion de *beacon* délimitant des supertrames. La structure de la supertrame est définie par les deux paramètres BI (*Beacon Interval*) qui définit l'intervalle qui sépare deux *beacons* consécutifs et SD (*Superframe Duration*) qui définit la durée de la supertrame.

BI et SD sont calculés en fonction de BO (*Beacon Order*) et SO (*Superframe Order*) selon les formules suivantes :

$$\begin{cases} BI = aBaseSuperframeDuration \cdot 2^{BO}, \\ SD = aBaseSuperframeDuration \cdot 2^{SO}, \end{cases}$$

avec  $0 \leq SO \leq BO \leq 14$ , *aBaseSuperframeDuration* est un attribut de la couche MAC qui définit la durée de la supertrame quand  $SO = 0$  (15.36 ms pour valeur par défaut pour *aBaseSuperframeDuration*).  $aBaseSuperframeDuration = aBaseSlotDuration * aNumSuperframeSlots$ , avec *aBaseSlotDuration* le nombre de symboles (un symbole vaut 4 bits) constituant un slot de la supertrame quand  $SO = 0$ . *aBaseSlotDuration* a 60 comme valeur par défaut. *aNumSuperframeSlots* est le nombre de slots qui constituent la supertrame, il vaut 16.

La portion active de la supertrame est découpée en *aNumSuperframeSlots* slots de temps égaux et est composée de trois parties : le beacon, la CAP (*Contention Access Period*) et la CFP (*Contention Free Period*) . Durant la CAP toutes les stations sont en compétition pour accéder au médium alors que la CFP est constituée de slots de temps, appelés GTS (*Guaranteed Time Slot*) alloués à chaque station pour communiquer avec le coordinateur.

Le début du premier slot est l'instant d'envoi du beacon. La CAP suit la fin de transmission du *beacon* et la CFP, si elle est présente, suit immédiatement la fin de la CAP. Si un coordinateur ne souhaite pas appliquer la structure de la supertrame, il initialise les valeurs de BO et de SO à 15. Dans ce cas, nous

nous retrouvons dans un PAN en mode non suivi de *beacon*.

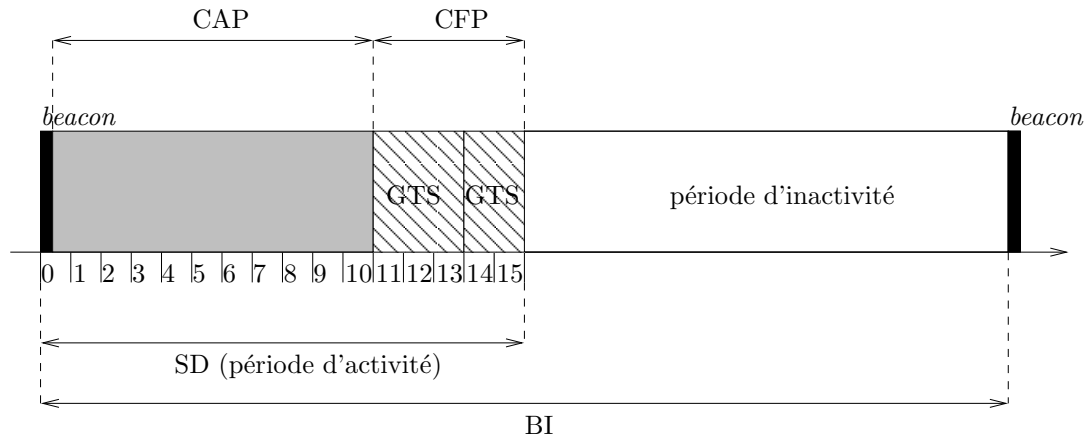


FIG. 2.16 – La structure de la supertrame.

La figure 2.16 montre un exemple d'une supertrame incluant un période de CAP, une période de CFP contenant deux GTS de tailles différentes et une période d'inactivité (car  $BO = SO + 1$ ).

L'accès au médium durant la CAP pour toute trame, sauf les acquittements et les *beacons*, est géré selon l'algorithme de CSMA/CA slotté. Avant tout envoi, l'entité doit s'assurer de pouvoir finir la transaction (incluant la réception d'un acquittement s'il est demandé) et de pouvoir attendre un IFS (*InterFrame Space*)<sup>7</sup> avant la fin de la CAP. Si ce n'est pas le cas, l'envoi est reporté à la CAP de la supertrame suivante. L'envoi de trames durant les slots réservés aux GTS s'effectue sans CSMA/CA.

**Algorithme de CSMA/CA slotté** L'algorithme de CSMA/CA slotté est appliqué pour l'envoi d'une trame durant la période CAP de la supertrame, sauf pour l'envoi des trames de *beacon* et des trames d'acquiescement. L'algorithme se base sur une unité de temps appelé période de *backoff*, une période de *backoff* est égale à  $aUnitBackoffPeriod$  symboles, soit 20 symboles. La synchronisation est basée sur un découpage du temps en intervalles à trois niveaux différents et imbriqués : des supertrames, elle-même découpées en slots, eux-même découpés en période de *backoff*. Les frontières des périodes de *backoff* de CSMA/CA pour chaque entité sont alignées avec les frontières des slots de la supertrame. Le début de la première période de *backoff* est le début du premier slot de la CAP (début de la transmission du beacon). Toute activité de la couche physique doit commencer à la frontière d'une période de *backoff*. Ainsi, toutes les entités sont synchronisées entre-elles sur les périodes de *backoff*.

La figure 2.17 montre la hiérarchie du découpage temporel et l'alignement des périodes de *backoff* avec les slots de la supertrame. Nous avons considéré le cas où  $BO = SO = 0$ , ce qui nous donne un slot de supertrame durant 0,96 ms. Dans ce cas, chaque slot de supertrame contient trois périodes de *backoff*.

<sup>7</sup>La durée d'un IFS dépend de la longueur du MPDU (*MAC Protocol Data Unit*) de la trame envoyée, ce qui correspond à la *payload* physique. Quand cette longueur dépasse 18 octets, le IFS vaut 40 symboles, sinon, le IFS vaut 12 symboles.

La supertrame montrée sur la figure 2.17 ne comporte pas de CFP.

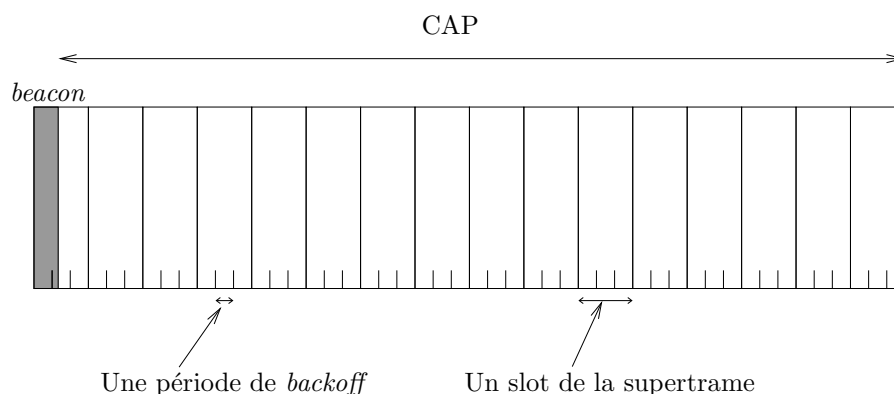


FIG. 2.17 – Les périodes de *backoff* et les 16 slots de la supertrame.

L'algorithme de CSMA/CA slotté est une variante de CSMA/CA qui exploite une synchronisation précise pour se dispenser de surveiller en permanence l'occupation du médium, ce qui consomme beaucoup d'énergie. Il est basé sur trois paramètres essentiels : NB, BE et CW. NB (*Number of Backoffs*) est le nombre de fois où l'entité a appliqué l'algorithme pour essayer d'envoyer la trame courante. BE (*Backoff Exponent*) est l'exposant de *backoff* qui définit la longueur de l'intervalle dans lequel il faut tirer un *backoff*. CW (*Contention Window*) définit le nombre de périodes de *backoff* consécutives à la fin desquelles le canal doit être détecté libre avant de commencer la transmission.

La figure 2.18 présente une version simplifiée des différentes étapes de l'algorithme de CSMA/CA slotté. La première étape est la phase d'initialisation. L'algorithme commence avec  $BE = macMinBE$ ,  $NB = 0$  et  $CW = 2$ . Une fois les paramètres initialisés, l'étape 2 consiste à localiser la frontière de la prochaine période de *backoff* et à attendre un nombre entier de périodes de *backoff* tiré aléatoirement dans l'intervalle  $[0; 2^{BE} - 1]$ . Si la durée du *backoff* tiré est plus longue que le nombre restant de périodes de *backoff* dans la CAP, l'algorithme consomme le *backoff* jusqu'à la fin de la CAP et reporte ce qui lui reste pour la CAP de la supertrame suivante.

Après la consommation de ce *backoff*, la couche MAC vérifie que le nombre restant de périodes de *backoff* dans la CAP est suffisant pour effectuer 2 CCA, transmettre la trame physique et recevoir un éventuel acquittement. Si c'est le cas, l'étape 3 est appliquée. Dans cette étape, la couche MAC demande à la couche PHY d'effectuer un CCA à la frontière de la prochaine période de *backoff*. Si le temps restant dans cette supertrame est suffisant, la couche MAC reporte le CCA au début de la CAP de la prochaine supertrame, en commençant avec un tirage de *backoff* pour éviter les collisions systématiques dues à plusieurs reports de CCA effectués par différentes entités [42].

Si la couche PHY détecte que le canal est libre, la couche MAC décrémente CW, puis teste si CW est nul. Si c'est le cas, la couche MAC demande à la couche PHY de commencer la transmission à la

frontière de la prochaine période de *backoff*. Si  $CW$  n'est pas nul, la couche MAC demande à la couche PHY de faire un autre CCA. Le médium n'est donc pas scruté en permanence, mais c'est deux CCA consécutifs positifs qui permettent de conclure que le médium est libre <sup>8</sup>.

Si la couche PHY détecte que le canal est occupé, la couche MAC incrémente  $NB$  et  $BE$  de 1 (à condition que  $BE$  reste inférieur ou égal à  $macMaxBE$ ) et remet  $CW$  à 2. Si  $NB = macMaxCSMABackoffs$ , l'algorithme renvoie un échec d'accès. C'est alors à la couche MAC de demander une retransmission de la même trame tant que le nombre de tentatives est plus petit que  $macMaxFrameRetries$  ( $macMaxFrameRetries = 3$  par défaut). Si  $NB < macMaxCSMABackoffs$ , l'algorithme retourne à l'étape 2.

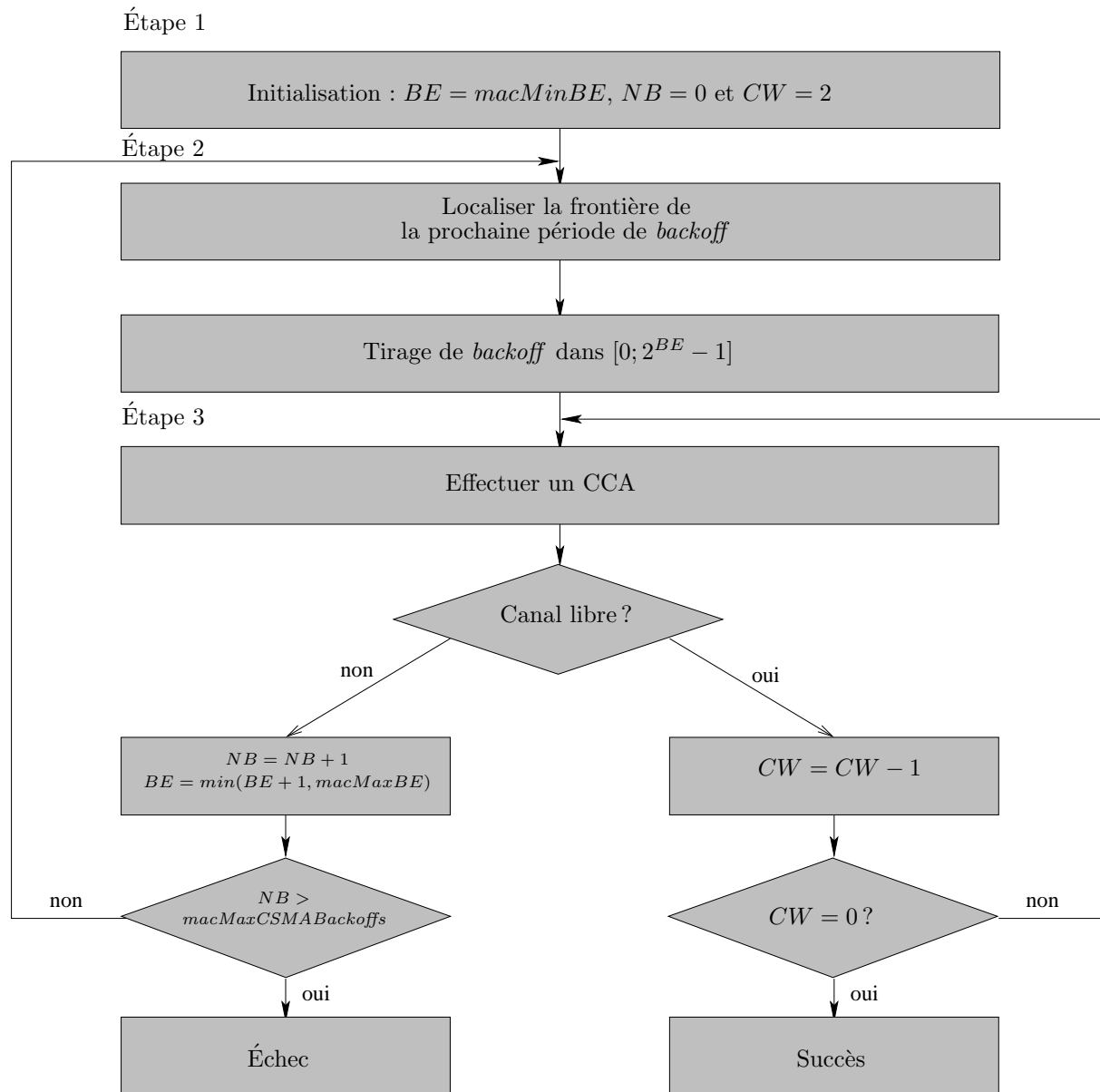


FIG. 2.18 – Diagramme de l'algorithme de CSMA/CA slotté de la norme IEEE 802.15.4.

<sup>8</sup>Dans le cas du CSMA/CA de la norme 802.11, la scrutation du canal est faite durant le backoff

**Le mode BLE (*Battery Life Extension*)** : le coordinateur de PAN peut décider de faire travailler le réseau en mode BLE pour économiser d'avantage de l'énergie. Un bit spécifique est réservé dans le *beacon* à cet effet. Quand ce bit est mis à 1, les entités associées à ce coordinateur doivent accéder au médium en respectant un IFS suivi d'une durée de *macBattLifeExtPeriods* après la réception du *beacon*.

*macBattLifeExtPeriods* est un entier qui correspond à un nombre de périodes de *backoff*. Il est calculé en fonction de trois termes : (i) la valeur maximum d'un *backoff* tiré dans une fenêtre avec  $BE = 2$ , cela vaut 3 périodes de *backoff*, (ii) la durée de CCA, ce qui fait 2 périodes de *backoff*, et (iii) la durée de transmission du préambule physique et du champ de synchronisation de l'en-tête physique, cela fait, pour la fréquence de 2.4 GHz, une longueur de 5 octets et donc une durée arrondie à une seule période de *backoff*.

Le total fait alors 6 périodes de *backoff*. Une entité qui souhaite émettre un message dans un réseau en mode BLE doit envoyer sa trame dans les 6 périodes de *backoff* qui suivent l'IFS après la réception du *beacon*. Le coordinateur et toutes les autres entités se mettent en mode sommeil si aucune trame n'est transmise avant cette durée.

**Algorithme de CSMA/CA non-slotté** L'algorithme de CSMA/CA non-slotté est appliqué pour les envois de trames dans un réseau en mode non suivi de *beacon*. Pour autant, il ne s'agit pas de CSMA/CA de la norme IEEE 802.11. Le paramètre *CW* n'existe pas dans l'algorithme de CSMA/CA non-slotté : il suffit de détecter une seule fois que le canal est libre pour commencer une transmission. L'unité du temps reste la période de *backoff*, mais l'activité des entités n'est pas synchronisée sur ces périodes de *backoff*.

La figure 2.19 présente les différentes étapes du CSMA/CA non-slotté. L'algorithme commence par la phase d'initialisation en mettant  $BE = macMinBE$  et  $NB = 0$ . Puis, elle tire aléatoirement un nombre entier de périodes de *backoff*. Après la consommation du *backoff*, la couche MAC demande à la couche PHY d'effectuer un CCA. Si la couche PHY détecte le canal libre, la couche MAC commence la transmission. Si ce n'est pas le cas, la couche MAC incrémente  $NB$  et  $BE$  (à condition que  $BE$  reste inférieur ou égal à  $macMaxBE$ ). Si  $NB < macMaxCSMABackoffs$ , l'algorithme revient au tirage de *backoff*. Si  $NB = macMaxCSMABackoffs$ , l'algorithme renvoie un diagnostic d'échec d'accès. C'est alors à la couche MAC de demander une retransmission de la même trame tant que le nombre de tentatives est plus petit que  $macMaxFrameRetries$  ( $macMaxFrameRetries$  vaut 3 par défaut).

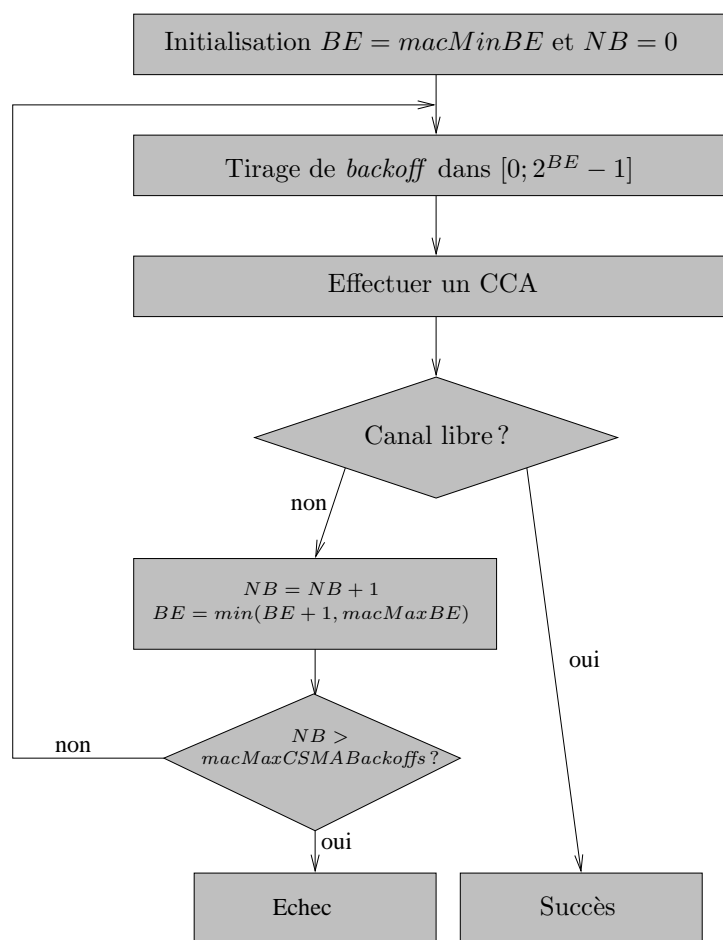


FIG. 2.19 – Diagramme de l'algorithme de CSMA/CA non-slotté de la norme IEEE 802.15.4.

### 2.2.2 Les *scans*, la création du réseau, les associations et la synchronisation

**Les *scans*** Afin de découvrir les réseaux existants à portée, de créer un réseau, de s'associer à un réseau ou de se synchroniser avec un réseau, la couche MAC effectue un des quatre types de *scans* suivants :

- Le *scan* d'énergie permet de récupérer l'énergie maximum détectée sur chaque canal. Ce scan est utilisé par un coordinateur souhaitant créer un PAN afin de choisir le canal convenable.
- Le *scan* actif permet de récupérer la liste des identifiants de PAN existants. Le scan actif suit la diffusion de requête de *beacon*. Un coordinateur fonctionnant selon un mode non suivi de *beacon* répond à cette requête par une diffusion de *beacon*. Un coordinateur fonctionnant selon un mode suivi de *beacon* l'ignore et continue à envoyer ses *beacons* périodiquement. Ce *scan* est utilisé par un coordinateur souhaitant créer un PAN pour choisir le bon identifiant de PAN, ou par une station qui cherche à s'associer à un PAN (dont elle connaît l'identifiant).
- Le *scan* passif comme pour le *scan* actif, permet de récupérer la liste des identifiants de PAN existants. En revanche, la station n'envoie pas une requête de *beacon* mais elle effectue une écoute passive de chaque canal à scanner pour une durée donnée. Ce scan est utilisé par une station qui cherche à s'associer à un PAN.
- Le *scan* d'orphelin permet à une station de retrouver son coordinateur après une perte de synchronisation avec ce dernier.

**Création du réseau** La couche supérieure envoie une requête à la couche MAC pour effectuer un *scan* actif sur une liste de canaux afin de découvrir les réseaux existants à portée. Une fois le bon canal choisi, la couche supérieure décide d'un identifiant de PAN et demande à la couche MAC d'initier un PAN avec cet identifiant.

**Association et désassociation** Après avoir effectué un *scan* (passif ou actif) et remonté le résultat dû à la couche supérieure, cette dernière demande à la couche MAC de s'associer à un PAN spécifique en précisant l'identifiant du PAN et l'adresse du coordinateur correspondant. Suite à cette demande, la couche MAC génère une requête d'association à destination du coordinateur.

À la réception d'une requête d'association, la couche MAC du coordinateur remonte l'information à la couche supérieure et c'est à celle-ci de décider d'accepter ou pas cette requête en fonction des ressources qu'il lui reste par exemple. La réponse à cette requête d'association est envoyée en mode de transmission indirecte (voir le paragraphe 2.2.3).

Suite à une requête de désassociation envoyée par la couche supérieure, la couche MAC envoie une indication de désassociation au coordinateur pour l'informer que la station souhaite se désassocier du PAN. De même, la couche supérieure d'un coordinateur peut décider de désassocier une station qui lui



est associée, elle lui envoie alors une commande de désassociation pour l'informer de ce fait.

**Synchronisation avec le *beacon* du coordinateur** Dans le mode suivi de *beacon*, les entités restent synchronisées au réseau grâce à la transmission périodique d'une trame de *beacon* par le coordinateur. Le tableau 2.4 montre les différents champs de ce beacon.

Octets : 2	1	2	2/8	0/5/6/ 10/14	2	variable	variable	variable	2
<i>Frame control</i>	Numéro de séquence	PAN ID source	Adresse source	Entête sécurité	Spécifications de la super-trame	Champs de GTS	Liste des adresses en attente	<i>Payload</i>	FCS
Entête MAC					<i>Payload</i> MAC				Footer MAC

TAB. 2.4 – Format du *beacon* de la norme IEEE 802.15.4.

Toute station appartenant à un *beacon-enabled* PAN doit pouvoir se synchroniser avec le *beacon* de son coordinateur pour connaître la structure de la supertrame, et aussi pour récupérer les trames la concernant (comme indiqué par la liste des entités ayant des trames en instance). Si une station ne reçoit pas un nombre *aMaxLostBeacons* consécutifs de *beacons*, la couche MAC détecte une perte de synchronisation et indique ce constat à la couche supérieure.

La couche MAC rejette toute trame de *beacon* dont l'adresse source ou l'identifiant du PAN ne correspondent pas à l'adresse du coordinateur ou à l'identifiant du PAN auquel la station est associée, respectivement.

### 2.2.3 Échange de données

Dans la primitive de requête de transmission de données envoyée par la couche supérieure, un champ (*txOptions*) spécifie dans quel mode de transmission des données doivent être transmises. Les trois modes d'échange sont : échange direct, échange indirect, échange en GTS. Ce champ spécifie aussi si la trame de données doit être acquittée ou pas.

**Échange direct** Durant la CAP, la station essaie d'envoyer la trame en CSMA/CA slotté. En cas de transmission avec demande d'acquittement, la couche MAC fait *macMaxFrameRetries* tentatives (par défaut 3 tentatives) si l'acquittement n'est pas reçu au bout de *macAckWaitDuration*.<sup>9</sup>

<sup>9</sup> $macAckWaitDuration = aUnitBackoffPeriod + aTurnaroundTime + phySHRDuration + 6 * phySymbolsPerOctet$ , où  $aUnitBackoffPeriod = 320 \mu s$ ,  $aTurnaroundTime = 192 \mu s$ ,  $phySHRDuration = 128 \mu s$  (pour la fréquence de 2.4 GHz), 6 représente le nombre d'octets de l'en-tête PHY incluant la longueur de la *payload* PHY de l'acquittement.

**Échange indirect** Pour favoriser l'aspect économie d'énergie, les stations feuilles initient la communication avec leur coordinateur, et cela selon deux procédures : (i) dans un PAN en mode suivi de beacon, le coordinateur indique dans son *beacon* la liste des adresses des entités qui ont des trames en attente chez lui, (ii) dans un PAN en mode non suivi de *beacon*, la station feuille sollicite le coordinateur pour vérifier s'il a des trames en attente pour elle.

Le coordinateur conserve les trames en attente pour une durée de *macTransactionPersistenceTime* avant de les supprimer si la station concernée ne l'a pas sollicité pour les récupérer.

**Échange en GTS** L'allocation des GTS est faite uniquement par le coordinateur du PAN. Un GTS est alloué soit en mode réception (du coordinateur du PAN vers la station) soit en mode émission (de la station vers le coordinateur du PAN).

Une station demande l'allocation d'un GTS auprès du coordinateur du PAN auquel elle est affiliée. Le nombre maximal de GTS est limité à 7, à condition que la durée de la CAP restante soit supérieure ou égale à *aMinCAPLength* (*aMinCAPLength* = 7 ms pour la fréquence de 2.4 GHz).

La réponse d'une requête de GTS est envoyée dans le *beacon* via des descripteurs de GTS. Cette information est gardée dans le *beacon* pour *aGTSDescPersistenceTime* (par défaut *aGTSDescPersistenceTime* = 4) *beacons* consécutifs.

La désallocation d'un GTS peut être faite soit à la demande de la couche supérieure de la station ayant le GTS, soit par la couche supérieure du coordinateur du PAN suite au constat que la station n'a plus utilisé son GTS durant  $2 * n$  supertrames consécutives<sup>10</sup>. Différents mécanismes de gestion d'allocation des GTS ont été proposés dans [43], [44] et [45].

## 2.3 Couche réseau ZigBee

La couche réseau de ZigBee est restée compatible avec la couche MAC IEEE 802.15.4 en assurant des fonctionnalités complémentaires et compatibles avec celles de la couche MAC. Il s'agit soit de fonctionnalités liées à la transmission de données, comme l'encapsulation des données applicatives et le choix du prochain saut du cheminement du paquet, soit de fonctionnalités liées à la gestion, comme la gestion des tables de routage, la configuration de la topologie et l'allocation des adresses logiques. Dans cette partie, nous allons détailler les fonctionnalités de gestion essentielles suivantes de la couche réseau de la pile IEEE 802.15.4/ZigBee :

- création de la topologie,
- allocation des adresses,
- routage.

---

<sup>10</sup> $n = 2^{8-BO}$  si  $0 \leq BO \leq 8$ ,  $n = 1$  si  $9 \leq BO \leq 14$ .

### 2.3.1 Création de la topologie

Comme pour la couche MAC IEEE 802.15.4, la couche réseau supporte deux types de topologies : topologie en étoile et topologie maillée. Un cas particulier d'une topologie maillée est la topologie arborescente. Le topologie arborescente est appelée *cluster-tree*.

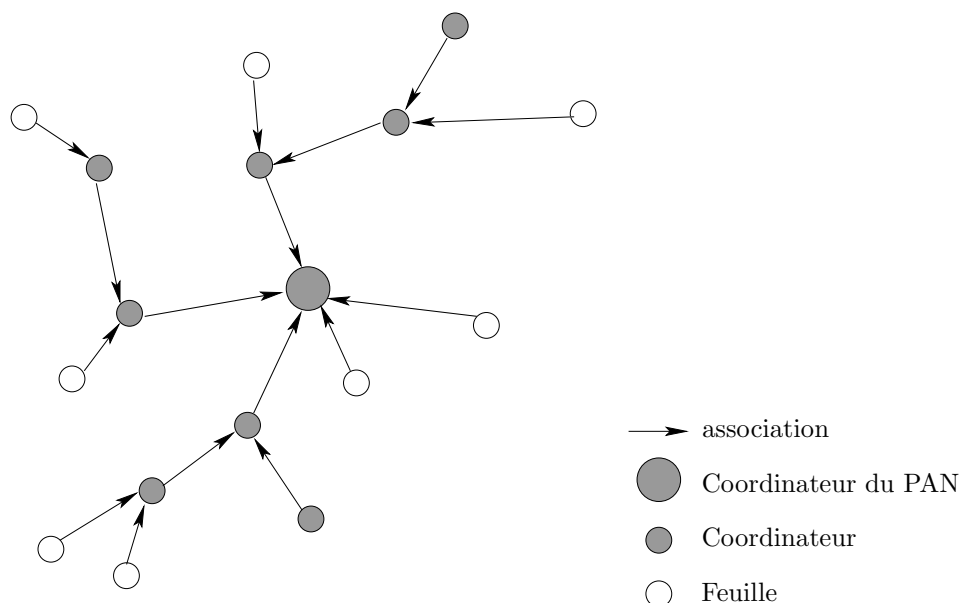


FIG. 2.20 – Topologie *cluster-tree*.

Dans un réseau de topologie *cluster-tree*, le coordinateur du PAN transmet dans le *beacon* trois paramètres essentiels que chaque coordinateur doit respecter et qui définissent la topologie en arbre du réseau. Ces trois paramètres sont :

- la profondeur maximale du réseau ( $Lm$ ),
- le nombre maximal de fils par coordinateur ( $Cm$ ),
- le nombre maximal de fils coordinateurs par coordinateur ( $Rm$ ) .

À noter que si  $Lm = 1$  nous nous retrouvons dans une topologie en étoile.

### 2.3.2 Allocation des adresses

Avant d'être associée au réseau, une entité n'a que son adresse MAC IEEE, appelée adresse longue. Cette adresse occupe 8 octets. Afin d'optimiser la taille des champs d'adressage dans les trames échangées, la couche réseau alloue une adresse courte logique qui n'occupe que 2 octets. Il existe deux mécanismes pour cette allocation : une allocation d'adresses hiérarchiques et une allocation d'adresses aléatoires.

**Allocation d'adresses hiérarchiques** L'allocation des adresses courtes se fait en se basant sur la profondeur du coordinateur et sur les trois paramètres  $Lm$ ,  $Rm$  et  $Cm$  [39]. Ce mécanisme d'allocation

est distribué. À chaque coordinateur est confié une plage d'adresses selon sa profondeur. L'étendue de cette plage d'adresses est appelée *Cskip*. *Cskip* pour une profondeur  $d$  donnée est calculé selon la formule suivante :

$$Cskip(d) = \begin{cases} 1 + Cm * (Lm - d - 1) & \text{si } Rm = 1, \\ \frac{1 + Cm - Rm - Cm * Rm^{Lm - d - 1}}{1 - Rm} & \text{sinon} \end{cases}$$

Les adresses sont allouées différemment selon le type de la station qui s'associe :

$$\begin{cases} Adr_C = Adr_P + 1 + nbC * Cskip(d), \\ Adr_F = Adr_P + Rm * Cskip(d) + n \end{cases}$$

où  $Adr_C$  désigne l'adresse allouée pour un nouveau coordinateur,  $Adr_P$  l'adresse du père,  $Adr_F$  l'adresse allouée pour une nouvelle feuille,  $nbC$  le nombre actuel de coordinateurs fils et  $n$  est un entier supérieur à 1 incrémenté suite à chaque nouvelle association d'une feuille ( $1 \leq n \leq (Cm - Rm)$ ).

Prenons le cas où  $Lm = 3$ ,  $Rm = 3$  et  $Cm = 5$ , valeurs que nous désignerons par (3, 3, 5). Cela nous donne les valeurs données sur le tableau 2.5 pour *Cskip* selon la profondeur  $d$  :

Profondeur	<i>Cskip</i>
0	21
1	6
2	1
3	0

TAB. 2.5 – Valeurs de *Cskip* pour l'exemple (3, 3, 5).

À noter qu'un coordinateur de profondeur  $Lm$  n'accepte aucune association. En appliquant le jeu de paramètres (3, 3, 5) au le *cluster-tree* de la figure 2.20, nous obtenons les adresses données sur la figure 2.21 pour chaque station.

**Allocation d'adresses aléatoires** Dans le mode d'allocation d'adresses aléatoires, les paramètres  $Lm$ ,  $Rm$ ,  $Cm$  et la profondeur n'ont plus d'effet sur le choix d'une adresse courte. Après avoir accepté l'association d'une entité, le coordinateur choisit aléatoirement une adresse courte, en vérifiant que cette adresse ne soit présente dans aucune entrée de sa table NIB (*Network layer Information Base*) . Les éventuelles conflits d'adresses sont détectés et résolus par un mécanisme supplémentaire [39].

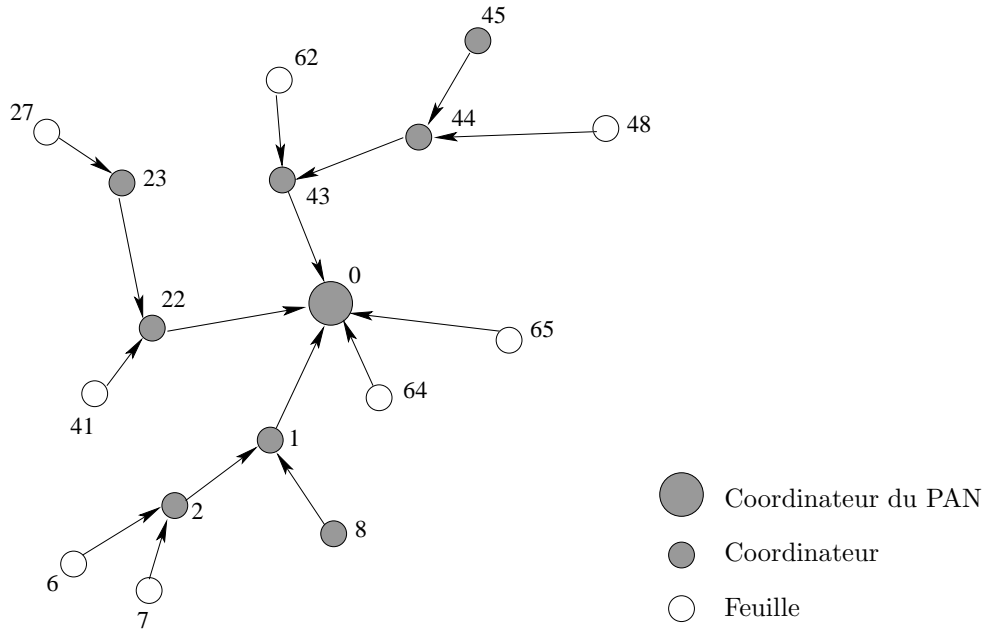


FIG. 2.21 – Allocation des adresses hiérarchiques sur un exemple. Le coordinateur 1 par exemple, qui est à la profondeur 1, a alloué l’adresse 2 à son premier fils et 8 au deuxième,  $8 = 2 + \text{Cskip}(1)$ .

### 2.3.3 Routage

Tout coordinateur qui possède des capacités de routage, maintient une table de routage qui sera utilisée pour déterminer le prochain saut pour les paquets qui sont à destination des stations qui se trouvent hors portée du coordinateur. Le protocole de routage utilisé est AODV (*Ad hoc On-demand Distance Vector*) [46]. Nous allons nous limiter à détailler le routage hiérarchique uniquement.

**Routage hiérarchique** L’allocation d’adresses hiérarchiques offre la possibilité d’appliquer un routage hiérarchique. Les coordinateurs n’ayant pas une stratégie de routage particulière appliquent le routage hiérarchique. Ce routage ne demande pas d’échanges supplémentaires pour trouver les routes, ni la construction de tables de routage.

Les routes utilisées pour acheminer les paquets dans un routage hiérarchique sont uniquement les routes de l’arbre. Pour déterminer le prochain saut dans un routage hiérarchique le coordinateur applique un algorithme qui retourne l’adresse du prochain saut en fonction de son adresse et l’adresse de la destination finale du paquet.

Nous désignons par  $Adr$  l’adresse du coordinateur qui cherche à router le paquet selon le routage hiérarchique,  $d$  sa profondeur,  $Adr_P$  l’adresse de son père,  $DestFinale$  l’adresse de la destination finale du paquet et  $Dest$  l’adresse du prochain saut.

$$\left\{ \begin{array}{ll} \text{si} & \text{Adr} < \text{DestFinale} \leq \text{Adr} + \text{Cskip}(d-1), \\ \\ \text{alors} & \text{si } \text{DestFinale} > \text{Adr} + \text{Rm} * \text{Cskip}(d), \\ \\ & \text{alors } \text{Dest} = \text{DestFinale}, \\ \\ & \text{sinon } \text{Dest} = \text{Adr} + 1 + \left\lfloor \frac{\text{DestFinale} - (\text{Adr} + 1)}{\text{Cskip}(d)} \right\rfloor * \text{Cskip}(d), \\ \\ \text{sinon} & \text{Dest} = \text{Adr}_P \end{array} \right.$$

Si  $\text{DestFinale}$  est comprise entre  $\text{Adr}$  et  $\text{Adr} + \text{Cskip}(d-1)$ , la destination finale est un descendant du coordinateur. Si ce n'est pas le cas, le coordinateur remonte le paquet à son père en mettant  $\text{Dest} = \text{Adr}_P$ .

Si la destination finale est un descendant, le coordinateur vérifie si elle est une de ses feuilles. Si c'est le cas, il met  $\text{Dest} = \text{DestFinale}$ . Si la destination finale n'est pas une feuille, pour trouver le fils vers lequel il va router le paquet, le coordinateur applique la formule  $\text{Dest} = \text{Adr} + 1 + \left\lfloor \frac{\text{DestFinale} - (\text{Adr} + 1)}{\text{Cskip}(d)} \right\rfloor * \text{Cskip}(d)$  qui retourne l'adresse du fils coordinateur auquel il faut envoyer le paquet.

## 2.4 Couche application ZigBee

La couche application de ZigBee est constituée de la sous-couche APL (*APplication sub-Layer*), du ZDO (*ZigBee Device Object*) et de l'*Application Framework* contenant des profils prédéfinis. La figure 2.22 représente les trois entités de la couche application.

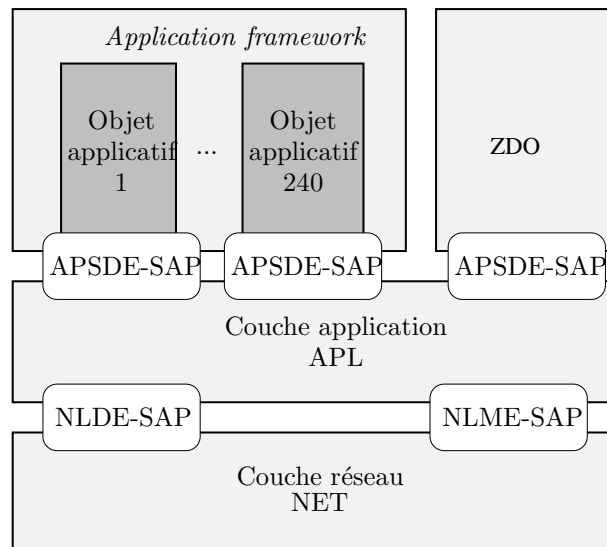


FIG. 2.22 – Couche application ZigBee. L'interface entre la couche application est les différents objets est appelée APSDE pour (*APplication Support sub-layer Data Entity*).

L'APS (*APplication Support*) constitue l'interface entre la couche réseau d'une part, le ZDO et les objets prédéfinis d'autre part. Nous listons à titre indicatif quelques services offerts par l'APS : l'encapsulation des données, la fragmentation et l'assemblage de données, le cryptage de données et l'adressage par groupe.

L'*Application Framework* peut supporter 240 objets applicatifs différents sur un même nœud ZigBee. L'objet applicatif par défaut est défini dans le ZDO. L'*Application Framework* permet aussi de définir des profils applicatifs spécifiant des formats de messages et la façon de les traiter entre un ensemble d'objets applicatifs appartenant à des nœuds ZigBee différents. La figure 2.23 représente un exemple de deux nœuds ZigBee A et B ayant chacun plusieurs objets applicatifs (interrupteur 1 et interrupteur 2 appartenant au nœud A et lampe 1, lampe 2, lampe 3 et lampe 4 appartenant au nœud B). Nous voyons comment l'objet applicatif Int1 (interrupteur 1) peut communiquer à plusieurs objets applicatifs (lampe 1, lampe 2 et lampe 3) grâce à une table de *binding* qui permet de formaliser les correspondances entre différents objets. Cette table est partagée par tous les nœuds qui partagent la même application. Les objets interrupteurs Int1 et Int2 et les lampes L1, L2, L3 et L4 appartiennent tous à un même profil applicatif, ce qui leur permet de comprendre les messages échangés entre-eux.

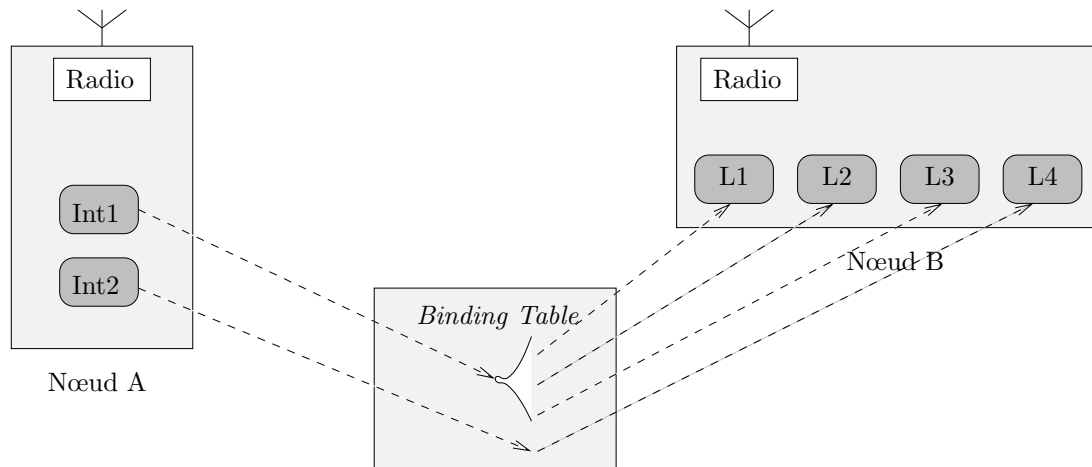


FIG. 2.23 – Exemples d'application et de profils ZigBee.

## 2.5 Problèmes persistants

Le groupe de travail IEEE 4b [47] a été créé dans le but d'affiner la norme IEEE 802.15.4 en trouvant des améliorations, résolvant des ambiguïtés, supprimant des complexités non nécessaires, rendant l'aspect sécurité plus flexible par exemple, et en traitant d'autres points concernant l'évolution de cette norme.

Ce groupe de travail a identifié le problème de collisions de *beacons* dont souffre la topologie *cluster-tree*, lorsque plusieurs coordinateurs envoient des *beacons*. Ils ont classé ces collisions selon deux types :

- les collisions directes,

- les collisions indirectes.

Le groupe 4b a aussi proposé des pistes de solutions à prendre en compte pour résoudre les problèmes identifiés, que nous allons présenter par la suite.

### 2.5.1 Collisions directes de *beacons*

La collision directe de *beacons* correspond au cas où deux ou plusieurs coordinateurs sont à portée les uns des autres et envoient leurs *beacons* en même temps. Cela est représenté sur la figure 2.24 où les coordinateurs C1 et C2 génèrent une collision directe de *beacons*.

Deux approches ont été proposées pour résoudre ce problème, l'approche *Beacon Only Period* et l'approche de supertrames asynchrones.

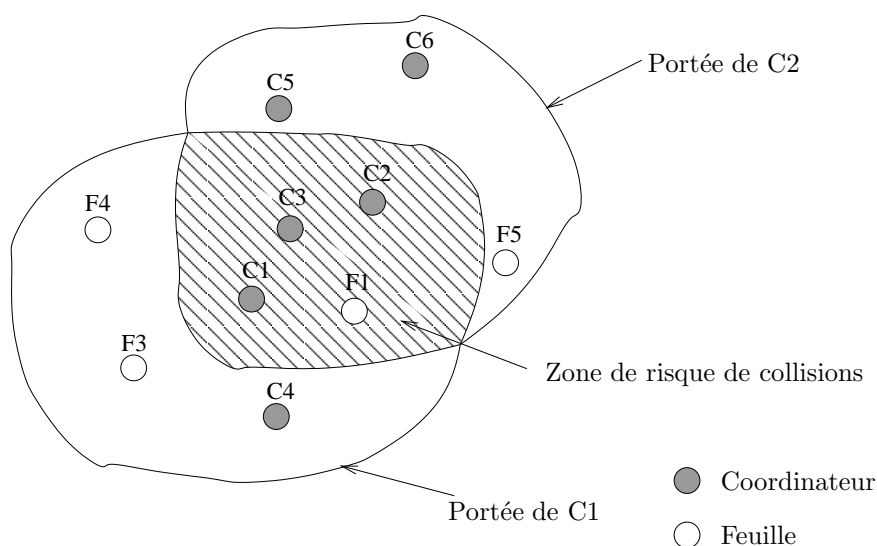


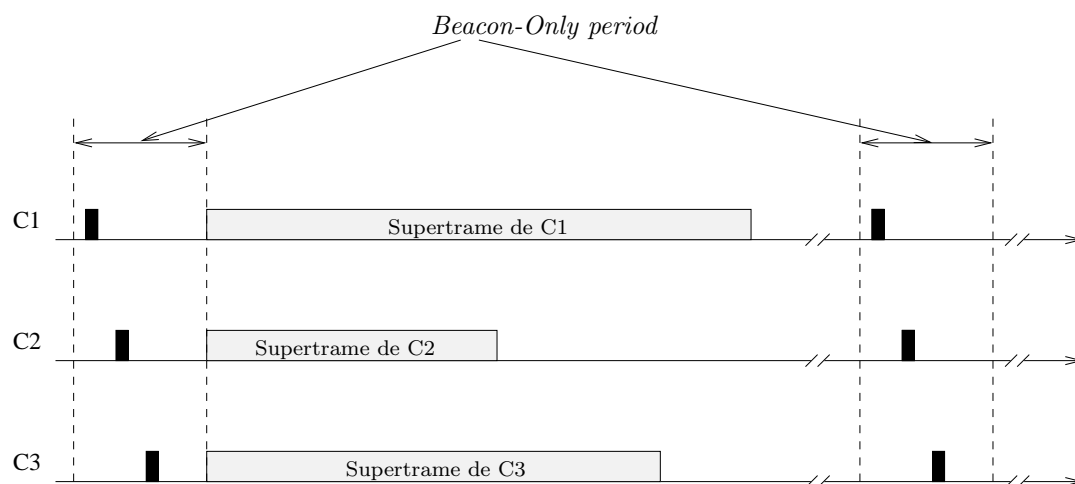
FIG. 2.24 – Collision directe de *beacons*.

**Approche *Beacon Only Period*** Dans cette approche, la solution proposée est de changer la structure de la supertrame en réservant une période en début de chaque supertrame. Durant cette période, tous les *beacons* sont envoyés comme le montre la figure 2.25. Chaque coordinateur maintient une table de coordinateurs voisins contenant l'instant d'envoi du *beacon* de chaque voisin. En se basant sur cette table, le coordinateur choisit un slot non utilisé par ses voisins pour envoyer son propre *beacon*.

Si deux coordinateurs, à portée l'un de l'autre, effectuent la même démarche, ils risquent de choisir le même slot libre. Cette solution ne résout donc pas complètement la collision directe de *beacon*.

Dans [48], les auteurs identifient le problème de la réutilisation d'un même slot pour envoyer le *beacon* qui cause la désynchronisation d'un père alors que ses fils se synchronisent avant lui. Pour résoudre ce conflit, les auteurs proposent (i) qu'aux coordinateurs fils soient alloués au moins un slot après le slot de leur père pour envoyer leurs *beacons* et (ii) que deux coordinateurs de la même profondeur dans l'arbre



FIG. 2.25 – Découpage temporel de l'approche *Beacon Only Period*.

qui sont suffisamment loin l'un de l'autre pour ne pas causer des collisions directes ou indirectes de *beacons* puissent diffuser leur *beacon* en même temps. Les auteurs ne précisent pas en revanche comment deux coordinateurs peuvent savoir qu'ils ne causent pas de collisions de *beacons*.

**Approche par supertrames asynchrones** Cette solution suppose que les supertrames des différents coordinateurs sont asynchrones, c'est à dire que les périodes d'activité des coordinateurs ne coïncident pas. La figure 2.26 montre une exemple de supertrames asynchrones de trois coordinateurs. Dans ce cas, chaque coordinateur choisit aléatoirement le début de sa supertrame dans les périodes d'inactivité de ses coordinateurs voisins.

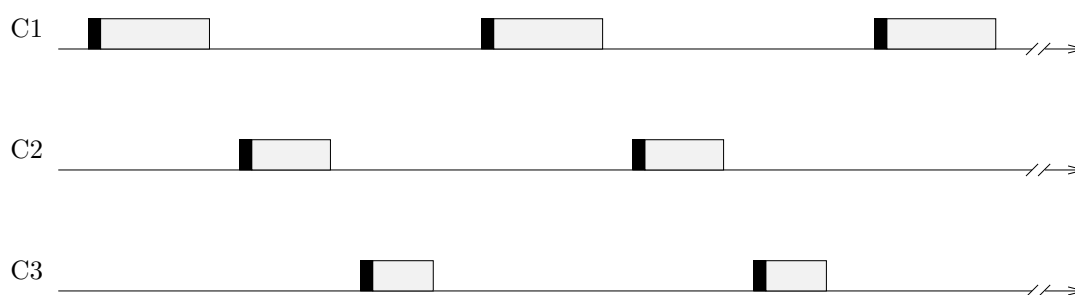


FIG. 2.26 – Découpage temporel de l'approche de supertrames asynchrones.

En appliquant cette approche, deux coordinateurs frères (tous les deux fils d'un même coordinateur) ne peuvent pas communiquer directement entre-eux, même s'ils sont à portée l'un de l'autre, puisqu'ils ne partagent pas le même intervalle de temps.

Une solution centralisée de cette approche est proposée dans [49]. Dans cette solution, le coordinateur du PAN indique dans son *beacon* l'intervalle de temps alloué à chaque coordinateur. Durant cet intervalle de temps, chaque coordinateur gère l'activité durant sa supertrame sans se préoccuper de celles de ses

coordinateurs voisins. L'algorithme d'allocation des intervalles de temps est détaillé mais les auteurs ne précisent pas comment l'information est diffusée dans le *beacon* ni comment le *beacon* est propagé pour atteindre tous les coordinateurs.

### 2.5.2 Collisions indirectes de *beacons*

La collision indirecte de *beacons* survient quand deux ou plusieurs coordinateurs, hors portée les uns des autres, envoient leurs *beacons* en même temps alors que sans être à portée les uns des autres ils couvrent les mêmes entités. Cela correspond au cas du terminal caché. Un exemple de collisions indirectes de *beacons*, entre les coordinateurs C1 et C2, est donné sur la figure 2.27. Deux approches ont été proposées pour éviter les collisions indirectes de *beacons* : l'approche réactive et l'approche proactive.

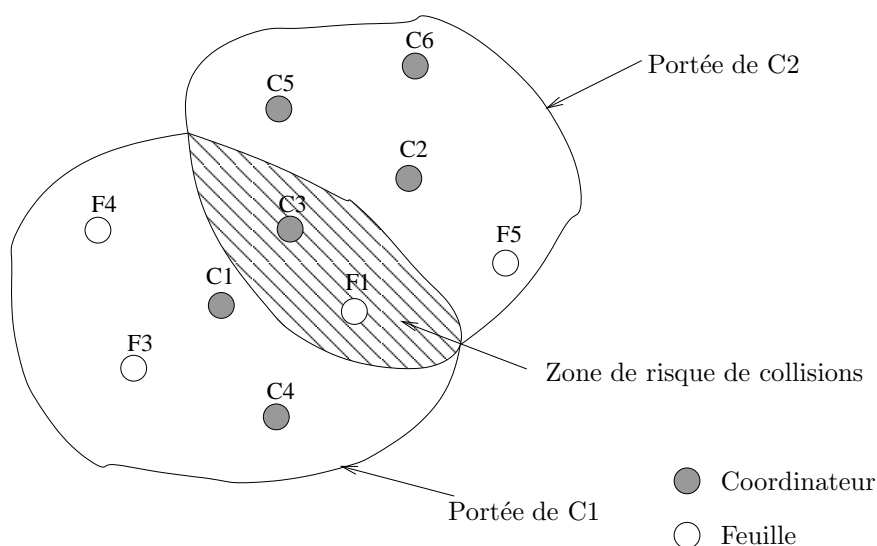


FIG. 2.27 – Collisions indirectes de *beacons*.

**Approche réactive** L'approche réactive consiste à résoudre le problème de collisions de *beacons* une fois détecté. Pour ce faire, la procédure de *scan* d'orphelinage doit être modifiée. L'entité qui détecte une perte de synchronisation diffuse une notification indiquant qu'elle est devenue orpheline. Quand un coordinateur reçoit cette notification, s'il se rend compte que l'entité lui est associée, il répond par un *coordinator realignment*. Sinon, il répond par un *beacon time notification* pour indiquer l'instant d'envoi de son *beacon*. Si l'entité qui a déclenché le *scan* d'orphelin ne reçoit ni un message de *coordinator realignment* ni un message de *beacon time notification*, elle constate qu'il y a pas d'entité à portée qui envoie des *beacons*. Si elle reçoit un des deux messages, l'entité analyse les informations reçues et diffuse un message de *beacon conflict notification* si elle détecte un conflit entre les instants d'envoi de *beacons*. À la réception d'un message de *beacon conflict notification*, les coordinateurs ajustent les instants d'envoi de leurs *beacons*.

**Approche proactive** L'approche proactive essaie de limiter les conflits de collisions indirectes de *beacons* durant la phase d'association. À la réception d'une requête d'association, toutes les entités (coordinateurs ou feuilles) répondent avec un message de *beacon time notification* pour indiquer l'instant d'envoi du *beacon* de leur père. De cette façon, un coordinateur reçoit des informations supplémentaires au moment de l'association pour limiter les risques de choisir le même instant d'envoi de *beacon* qu'un autre coordinateur avec qui il pourrait avoir des conflits.

## 2.6 Synthèse

D'après les différents protocoles étudiés, il s'avère que la création de clusters est une technique efficace, elle d'ailleurs adoptée par la norme IEEE 802.15.4/ZigBee pour économiser de l'énergie mais aussi par d'autres protocoles de la couche réseau [50, 51]. Dans la suite, nous allons proposer une méthode basée sur la même technique de création de *cluster* dérivée de la norme IEEE 802.15.4/ZigBee. Dans cette étude, nous considérons que les *clusters* existent de fait, leur existence sera exploitée pour leur allouer des intervalles de temps propres de façon à éviter qu'ils soient localement en concurrence pour accéder au médium.

## Chapitre 3

# Notre contribution

Dans ce chapitre, nous présentons notre contribution qui est le protocole MaCARI. MaCARI est un protocole MAC qui découpe le temps en cycles globaux. Chaque cycle global est segmenté temporellement en plusieurs périodes séquentielles pour améliorer l'utilisation du médium. Dans la première partie, nous décrivons d'une manière générale les fonctionnalités de MaCARI. Dans la deuxième partie, nous détaillons comment la synchronisation sur les différentes périodes se fait et nous étudions les différentes façons d'amélioration de cette période. Dans la troisième partie, nous expliquons comment la segmentation temporelle est réalisée.

## 1 MaCARI

MaCARI (protocole MAC pour OCARI) est un protocole MAC économe en énergie qui offre une différenciation de service selon le type de trafic. L'économie d'énergie est obtenue en faisant dormir les nœuds alors que la différenciation de service est réalisée en faisant du relais de trames ou du routage de paquets selon la priorité du trafic. La mise en veille des nœuds exige une synchronisation des périodes d'activité et des périodes d'inactivité afin d'éviter d'avoir des dégradations de performances du réseau dues à des déphasages de cycles d'activité. Cette synchronisation est réalisée en ayant un nœud qui initie une diffusion d'un *beacon* qui contient les informations temporelles nécessaires pour tous les nœuds du réseau. En s'appuyant sur cette synchronisation, MaCARI découpe le temps en différentes périodes, chacune ayant des caractéristiques propres.

Dans la suite de cette partie, nous décrivons les fonctionnalités essentielles de MaCARI puis les échanges avec la couche supérieure permettant la prise en compte de la différenciation de service.

## 1.1 Description générale de MaCARI

La couche MaCARI est la couche MAC de la pile OCARI. Parmi ses fonctionnalités essentielles, nous retrouvons les fonctionnalités traditionnelles d'une couche MAC comme l'encapsulation de trames et l'accès au médium, et des fonctionnalités supplémentaires comme la gestion de la topologie et l'affectation des adresses hiérarchiques. La liste des fonctionnalités est la suivante :

- gestion de la synchronisation et du découpage temporel de l'activité du réseau,
- encapsulation niveau 2 des trames,
- estampillage temporel des trames de données,
- accès au médium,
- création du réseau et gestion de sa topologie,
- affectation des adresses hiérarchiques,
- relais garanti de trames pour une différenciation de service.

Dans le réseau OCARI, il existe trois types de nœuds : un coordinateur du PAN, des coordinateurs et des feuilles.

**Le coordinateur du PAN**, appelé CPAN par la suite, est le nœud qui crée le réseau, initie et maintient une synchronisation entre l'ensemble des nœuds du réseau. Il joue aussi le rôle de puits de données et passerelle pour interconnecter son îlot avec le reste du réseau OCARI. En dehors de ces rôles, le CPAN agit comme un coordinateur.

**Les coordinateurs** sont des nœuds qui participent au routage des informations. De plus, ils sont chargés de propager le *beacon* initié par le CPAN et de permettre à de nouvelles entités de rejoindre le réseau.

**Les feuilles** sont les capteurs et/ou les actionneurs ayant un rôle applicatif à remplir. Les feuilles sont attachées à un coordinateur et communiquent uniquement avec lui.

Un coordinateur et l'ensemble des feuilles qui lui sont associées constituent une étoile. Un exemple d'une topologie d'un îlot du réseau OCARI est présenté sur la figure 3.1. Dans la suite de cette partie, nous détaillons chacune des fonctionnalités essentielles de MaCARI.

Dans la suite, nous détaillons chacune des fonctionnalités essentielles de MaCARI.

### 1.1.1 Synchronisation et découpage temporel

MaCARI divise le temps en trois périodes principales qui forment un cycle global :

- une période de synchronisation  $[T_0; T_1]$ ,
- une période d'activité  $[T_1; T_3]$  (elle-même sous-découpée en  $[T_1; T_2]$  et  $[T_2; T_3]$ ),
- une période d'inactivité  $[T_3; T_0]$ .

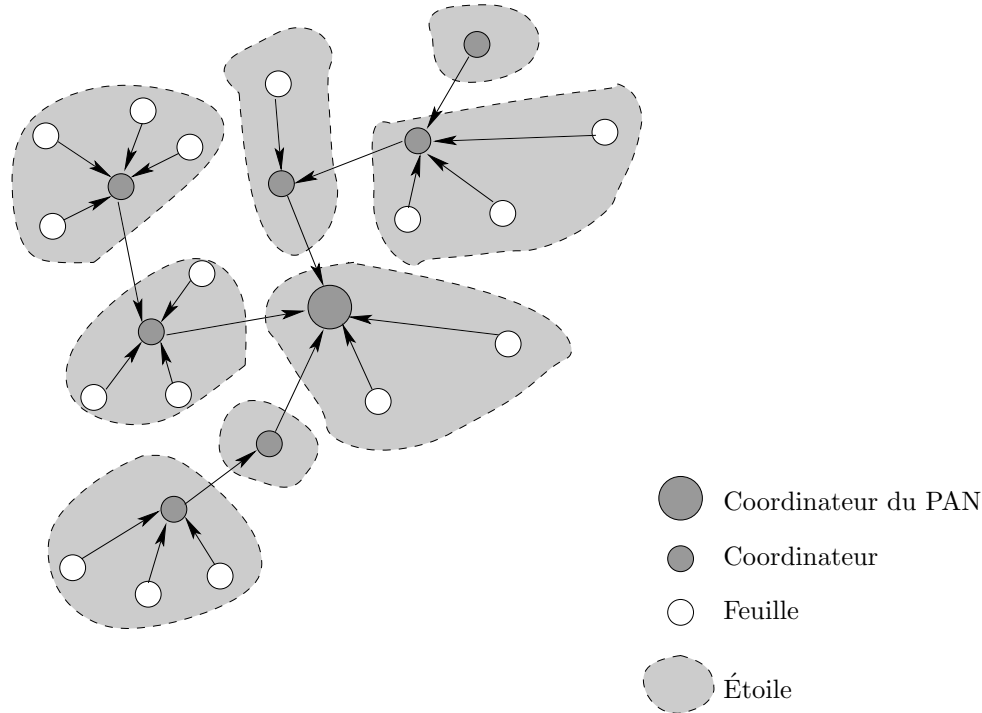


FIG. 3.1 – Exemple d’une topologie d’un réseau OCARI.

Ces périodes sont présentées sur la figure 3.2.

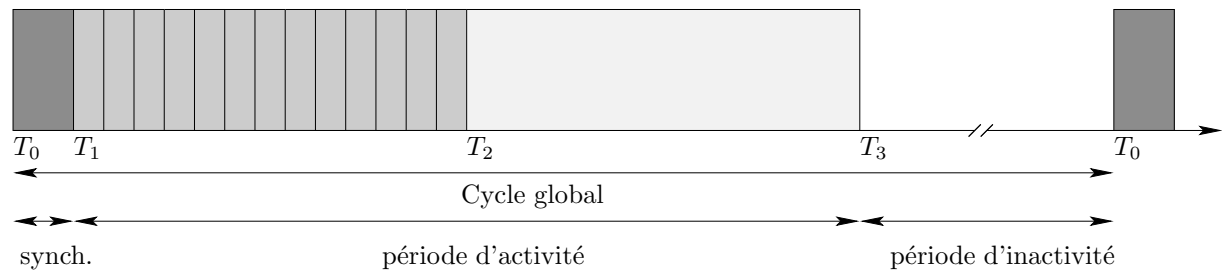


FIG. 3.2 – Le cycle global dans MaCARI.

Durant la période de synchronisation, le coordinateur du PAN diffuse un *beacon* de synchronisation qui précise le dimensionnement du cycle global ainsi que le sous-découpage de la période d’activité. Pour pouvoir atteindre toutes les stations du réseau qui ne sont pas forcément à portée du CPAN, les coordinateurs propagent, à tour de rôle, le *beacon* diffusé par le CPAN. L’ordre dans lequel les coordinateurs doivent propager le *beacon* est porté par ce dernier.

La période d’activité est décomposée en deux parties : (i) une période de séquençement d’activités et de relais et (ii) une période de routage. Le séquençement d’activités et le relais se font dans  $[T_1 ; T_2]$  où à chaque étoile est alloué un intervalle de temps propre durant lequel le coordinateur de l’étoile communique avec les feuilles qui lui sont associées. Nous appelons cet intervalle de temps la période d’activité d’une étoile. Cet intervalle de temps permet à chaque étoile d’organiser son activité sans

craindre des collisions ou des interférences dues à l'activité d'autres étoiles en un même lieu. Ainsi, un coordinateur est capable d'allouer un GTS (selon la norme IEEE 802.15.4) sans craindre que les slots soient utilisés par le coordinateur d'une autre étoile. Un autre intervalle de temps supplémentaire est aussi alloué à chaque coordinateur (sauf au CPAN), durant lequel il communique avec son père. Nous appelons cet intervalle de temps l'intervalle de relais. Ces intervalles de relais sont prévus pour les échanges de trames protégées qui doivent être envoyées sans risque de collisions. Ce sont en fait des intervalles de temps à accès garanti étant donné que le coordinateur et son père sont les deux seules stations actives en ce lieu dans le réseau, durant cet intervalle de temps. Ce découpage temporel est représenté sur la figure 3.3.

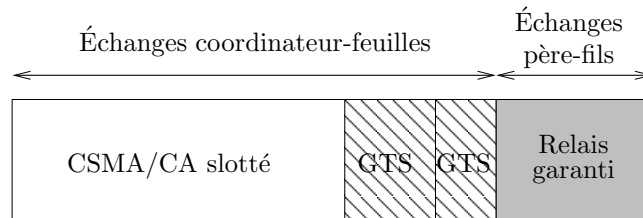


FIG. 3.3 – Une période d'activité d'une étoile suivie par un intervalle de relais.

À  $T_2$  toutes les étoiles ont eu une période d'activité qui a permis aux coordinateurs de récolter le trafic généré par leurs feuilles et relayer la partie prioritaire de ce trafic à leurs pères ou leurs fils. L'acheminement de la partie du trafic non-prioritaire se fait dans  $[T_2; T_3]$ . Seuls les coordinateurs sont actifs durant cette période<sup>1</sup>. Ils communiquent entre eux selon un protocole de routage hiérarchique modifié basé sur la table de voisinage, ce protocole est détaillé dans 1.1.4.

Durant la période d'inactivité, toutes les entités du réseau économisent de l'énergie en se mettant en mode sommeil. La durée de cette période dépend de la réactivité exigée par l'application. Se pose ici le problème du compromis entre l'économie d'énergie et le délai (que ce soit le délai d'accès au médium ou le délai de bout-en-bout). Plus cette période est longue, plus on économise de l'énergie mais plus le délai introduit est grand.

L'activité d'un nœud dans le réseau OCARI dépend de son type. Un coordinateur est actif durant la période  $[T_0; T_1]$ , de  $T_0$  jusqu'à ce qu'il envoie son *beacon*, alors qu'une feuille est active de  $T_0$  jusqu'à ce qu'elle reçoive un *beacon*. Dans  $[T_1; T_2]$ , un coordinateur est actif durant la période d'activité de son étoile, l'intervalle de relais avec son père et durant les intervalles de relais de ses fils. Une feuille est active uniquement durant la période d'activité de son étoile. Dans  $[T_2; T_3]$ , tous les coordinateurs sont actifs pour acheminer les paquets à leur destination finale, les feuilles ne sont pas actives. Finalement, aucune station n'est active durant  $[T_3; T_0]$ .

<sup>1</sup>Pour cette partie, les choix faits pour cette thèse diffèrent des travaux faits dans le cadre d'OCARI par notre partenaire de l'INRIA.

### 1.1.2 Encapsulation des trames

MaCARI assure son rôle de couche MAC et encapsule les trames niveau MAC avec un entête et un *footer*. L'en-tête contient des informations concernant le type de la trame, le type et le priorité du trafic, la nécessité d'un acquittement ou pas, les adresses source et destination, et le numéro de séquence. Le *footer* contient 16 bits pour effectuer une vérification de l'intégrité des trames ou FCS (*Frame Check Sequence*) basé sur un contrôle de redondance cyclique ou CRC (*Cyclic Redundancy Check*) permettant de détecter les erreurs de transmission avec une certaine probabilité de réussite. Le format général d'une trame MaCARI est spécifié dans le tableau 3.1.

<b>Octets :</b> <b>2</b>	<b>1</b>	<b>0/2</b>	<b>0/2</b>	<b>0/2/8</b>	<b>0/2</b>	<b>0/2/8</b>	<b>variable</b> <b>(114</b> <b>max)</b>	<b>2</b>
<i>Frame control</i>	<i>Sequence number</i>	<i>Dest. PAN ID</i>	<i>Dest. address</i>	<i>Final dest. address</i>	<i>Src. PAN ID</i>	<i>Src. address</i>	<i>Payload</i>	FCS
<b>MaCARI header</b>							<b>MaCARI payload</b>	<b>MaCARI footer</b>

TAB. 3.1 – Format général d'une trame MaCARI.

Plus de détails sur les champs d'une trame MaCARI sont donnés dans l'annexe B.

### 1.1.3 Estampillage temporel des trames

Grâce à la synchronisation sur l'instant  $T_1$  commun à toutes les stations, MaCARI possède une connaissance temporelle qui lui permet de fournir une notion de temps partagée par toutes les stations. Ce temps peut servir à mettre en œuvre un mécanisme d'estampillage pour les trames de données. Cette estampille peut être exploitée par la couche applicative pour dater les données qu'elle reçoit, plus de détails seront donnés en 2.5.

### 1.1.4 Accès au médium

L'accès au médium diffère dans MaCARI selon les intervalles de temps. Les *beacons* dans  $[T_0 ; T_1]$  sont envoyés directement sans CSMA/CA. Les échanges intra-étoile sont gérés en CSMA/CA slotté pour le trafic de type non-prioritaire et sans CSMA/CA durant les intervalles de GTS pour le trafic de type prioritaire. Les échanges durant les intervalles de relais garanti sont effectués sans CSMA/CA. Durant la période  $[T_2 ; T_3]$ , l'accès est géré selon le protocole CSMA/CA slotté décrit dans 2.2.



### 1.1.5 Création du réseau et gestion de la topologie

Le CPAN est le nœud responsable de la création du réseau. Il choisit un identifiant pour le PAN (nommé PAN ID) et diffuse un *beacon* périodiquement. Les entités qui reçoivent ce *beacon* peuvent envoyer une requête d'association au CPAN pour rejoindre le réseau, soit en tant que coordinateur, soit en tant que feuille. Une fois accepté en tant que coordinateur, le nouveau nœud commence à propager le *beacon* et permet à d'autres entités de rejoindre le réseau. Après chaque association d'un nouveau coordinateur, une primitive est remontée au CPAN pour l'informer de cette nouvelle association. Ainsi, une topologie arborescente est créée entre les différents nœuds du réseau. À noter que MaCARI utilise les trois paramètres topologiques  $Rm$ ,  $Lm$  et  $Cm$  détaillés dans 2.3.1 du chapitre 2.

### 1.1.6 Affectation des adresses hiérarchiques

Le CPAN possède toujours l'adresse 0. Au moment de l'association, à chaque nœud est affectée une adresse hiérarchique selon les règles de ZigBee décrites dans 2.3.2 du chapitre 2. Cela permet à MaCARI d'appliquer le protocole de routage hiérarchique décrit dans 2.3.3 du chapitre 2.

### 1.1.7 Relais garanti pour une différenciation de service

Dans le réseau OCARI, un relais de trames niveau MAC est effectué par MaCARI, pour faire cheminer le trafic de type prioritaire. Ayant connaissance de l'adresse hiérarchique de la destination finale et des paramètres topologiques ( $Lm$ ,  $Rm$  et  $Cm$ ), MaCARI réalise ce relais en suivant l'arbre topologique et en appliquant le protocole de routage hiérarchique, décrit dans 2.3.3 du chapitre 2, durant les intervalles de relais garanti entre père et fils sans passer par la couche réseau, ceci sans solliciter la couche réseau des stations intermédiaires du chemin suivi pas les trames.

Ainsi, MaCARI est capable d'offrir deux niveaux de qualités de service différents : un accès garanti pour un trafic prioritaire et un accès non-garanti pour un trafic non-prioritaire. L'accès garanti est réalisé durant les intervalles de relais garanti entre les couples de coordinateurs père-fils, et durant les GTS réservés aux communications entre le coordinateur et ses feuilles. L'accès non-garanti est réalisé durant les périodes d'activité des étoiles et durant  $[T_2; T_3]$  en utilisant l'algorithme de CSMA/CA slotté.

## 1.2 La pile OCARI

Dans le cadre du projet OCARI, nous avons participé à la conception d'une pile protocolaire dont MaCARI constitue la couche MAC. Cette pile est présentée sur la figure 3.4. La couche physique est celle de la norme IEEE 802.15.4. La couche réseau est constituée principalement de deux protocoles : EOLSR comme protocole de routage économe en énergie et SERENA comme protocole de coloriage permettant

d'activer/désactiver les nœuds pendant  $[T_2; T_3]$  pour économiser de l'énergie et pour optimiser l'accès au canal.

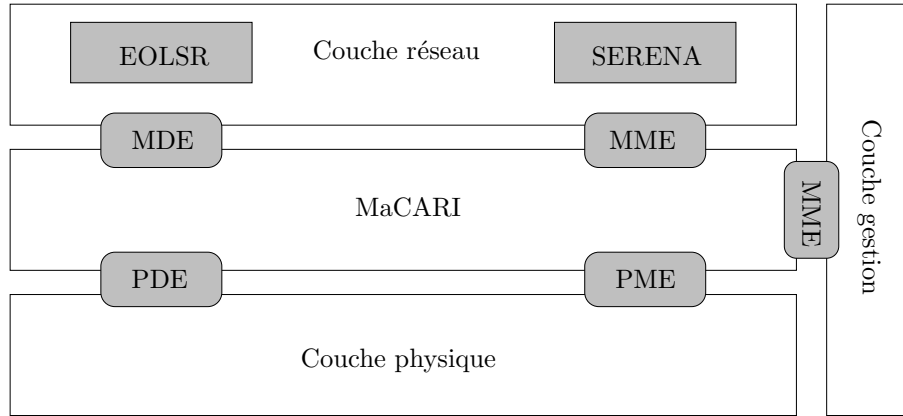


FIG. 3.4 – La pile OCARI.

Les communications entre les couches se font en utilisant des primitives. Dans la suite du document, nous présentons quelques primitives essentielles qui montrent l'interaction et l'aspect *cross-layering* entre les couches. Vous trouvez la liste complète des primitives de la couche MaCARI dans l'annexe C.

La figure 3.4 montre les différentes interfaces de communication entre MaCARI et les couches adjacentes. Les interfaces inter-couches sont :

- PDE (*Physical Data Entity*) pour les échanges des primitives de données entre MaCARI et la couche physique,
- MDE (*MAC Data Entity*) pour les échanges des primitives de données entre MaCARI et la couche réseau,
- MME (*MAC Management Entity*) pour les échanges des primitives de contrôle entre MaCARI d'un côté et les couches gestion et réseau d'un autre côté,
- PME (*Physical Management Entity*) pour les échanges des primitives de contrôle entre MaCARI et la couche physique.

### 1.2.1 EOLSR

Dans le cadre du projet OCARI, le protocole de routage a été traité par le partenaire INRIA avec lequel nous avons grandement collaborés compte tenu des grandes interactions entre les couches MAC et réseau.

EOLSR (*Energy-aware OLSR*) [52] est une version modifiée de OLSR (*Optimized Link State Routing*) [53] qui prend en compte l'énergie résiduelle des nœuds. OLSR est un protocole de routage proactif pour lequel les nœuds échangent périodiquement des informations topologiques pour construire des routes vers tous les nœuds du réseau. OLSR se base sur le principe de relais multi-points MRP (*MultiPoint Relay*)

où chaque nœud choisit (indépendamment des autres nœuds) des MPR pour atteindre l'ensemble de son voisinage à deux sauts.

EOLSR choisit les MPR (rebaptisé EMPR (*Energy efficient MPR*)) selon leur énergie résiduelle. Par la suite, les routes sont construites uniquement avec des nœuds EMPR.

**Interaction MaCARI-EOLSR** MaCARI se charge de remonter à EOLSR une indication de rupture de liens.

### 1.2.2 SERENA

SERENA (*SchEdule RoutEr Nodes Activity*) [54] est un algorithme qui effectue un séquençement des activités des coordinateurs pour leur permettre de travailler en même temps quand la topologie du réseau le permet, de dormir et ainsi d'économiser de l'énergie, tout en garantissant des communications de bout-en-bout. C'est un algorithme décentralisé d'allocation de slots de temps qui est exécuté localement dans chaque nœud. Chaque nœud possède une couleur et à chaque couleur est associé un slot de temps. Un nœud reste actif durant le slot de temps de sa couleur et durant les slots de temps des couleurs de ses voisins à un saut.

SERENA est décomposé en deux algorithmes essentiels : un algorithme de coloriage qui interdit l'utilisation de la même couleur dans un voisinage à trois sauts, et un algorithme d'allocation de slots de temps qui alloue un slot de temps pour chaque couleur. Les slots de temps sont utilisés pour l'envoi des paquets.

Le coloriage à trois sauts est nécessaire pour permettre l'envoi des acquittements MAC suite à la réception des trames envoyées en mode *unicast*. Dans la figure 3.5, le nœud D ne peut pas avoir la même couleur que A parce que l'acquittement envoyé par C (suite à l'envoi d'une trame de données de D pour C) risque de causer une collision auprès de B qui est censé recevoir les trames envoyées par A.

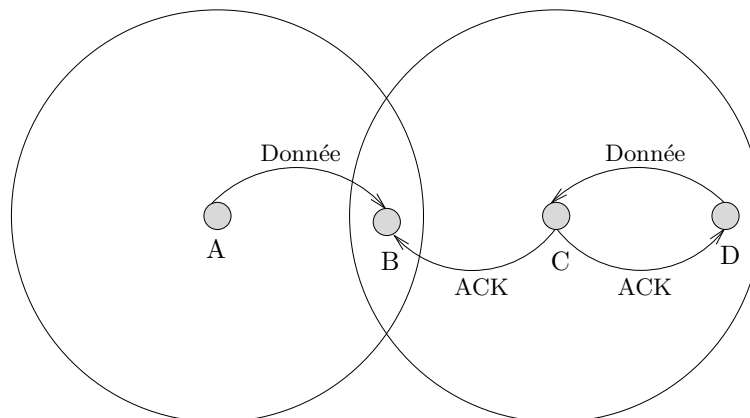


FIG. 3.5 – Nécessité d'un coloriage à trois sauts.

Dans le cadre d'OCARI seuls les coordinateurs ont une capacité de routage, ainsi SERENA est appliqué uniquement par les coordinateurs durant  $[T_2; T_3]$ .

**Interaction MaCARI-SERENA** La partie allocation des slots de temps de SERENA est confiée à MaCARI. SERENA se charge d'acheminer vers le CPAN la liste des couleurs utilisées dans le réseau. Par la suite, le CPAN annonce le découpage temporel et l'intervalle de temps alloué à chaque couleur dans le *beacon* de synchronisation.

La figure 3.6 montre un exemple de séquençement des slots colorés avec l'algorithme de SERENA dans  $[T_2; T_3]$ .  $Coul_1$ ,  $Coul_2$ ,  $Coul_3$ ,  $Coul_4$  et  $Coul_5$  représentent un exemple de cinq couleurs utilisées par SERENA. Dans la mise en œuvre retenue, les intervalles de temps sont égaux et séquencés à partir de  $T_2$ . Le reste de la période  $[T_2; T_3]$  reste non coloré pour que les nouveaux arrivants dans le réseau puissent router avant d'avoir obtenu une couleur.

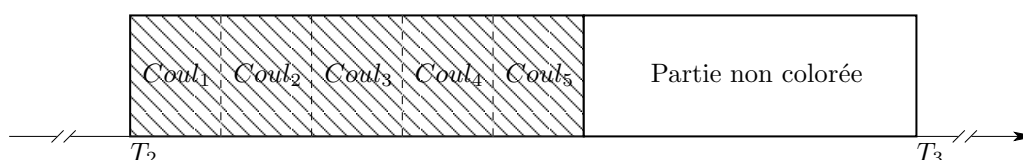


FIG. 3.6 – Séquençement des slots colorés avec l'algorithme de SERENA dans  $[T_2; T_3]$ .

### 1.2.3 Les échanges de primitives de données entre la couche supérieure et MaCARI

La couche supérieure transmet à MaCARI des requêtes d'envoi de données en spécifiant leur type : soit des données de type prioritaire, soit des données de type non-prioritaire. MaCARI gère l'accès garanti à deux niveaux : les communications entre coordinateurs durant les intervalles de relais garanti et les communications entre coordinateurs et leurs feuilles *via* le mécanisme des GTS. Il faut considérer quatre cas de figure :

**Données de type prioritaire dans le cas d'un coordinateur** : si la destination finale est une feuille associée au coordinateur, MaCARI vérifie si la feuille possède un GTS en mode réception et lui communique la trame durant le GTS. Si la feuille ne possède pas de GTS, la trame est envoyée en CSMA/CA slotté durant l'intervalle intra-étoile. Si la destination finale est un coordinateur ou une feuille appartenant à un autre coordinateur, MaCARI ajoute la trame à la file d'attente des trames à envoyer durant l'intervalle de relais garanti. Le prochain saut de cette trame est calculé par MaCARI au moyen du protocole de routage hiérarchique détaillé dans 2.3.3 du chapitre 2,

**Données de type non-prioritaire dans le cas d'un coordinateur** : si le prochain saut est un coordinateur<sup>2</sup>, MaCARI ajoute la trame à la file d'attente des trames à envoyer durant l'intervalle de

<sup>2</sup>Ce prochain saut n'est pas calculé par MaCARI, mais par le protocole de routage de la couche réseau.

temps  $[T_2 ; T_3]$  en CSMA/CA slotté. Sinon, la trame est envoyée en CSMA/CA slotté durant l'intervalle intra-étoile.

**Données de type prioritaire dans le cas d'une feuille :** MaCARI vérifie d'abord si la feuille possède un GTS en mode envoi. Si c'est le cas, la trame est envoyée durant ce GTS. Si ce n'est pas le cas, MaCARI remonte une erreur à la couche supérieure. C'est à la couche supérieure de prendre l'initiative de demander à MaCARI d'envoyer une requête de GTS (en mode envoi)<sup>3</sup> au coordinateur de la feuille.

**Données de type non-prioritaire dans le cas d'une feuille :** MaCARI envoie la trame en mode CSMA/CA slotté durant l'intervalle de temps intra-étoile.

---

<sup>3</sup>Les GTS sont réservés soit en mode envoi pour qu'une feuille puisse envoyer des données à son coordinateur, soit en mode réception pour que le coordinateur puisse envoyer des données à la feuille.

## 2 La période de synchronisation

MaCARI est un protocole économe en énergie, pour cela il met les nœuds en mode sommeil ceci permet de prolonger la durée de vie du réseau. Une étape de synchronisation est indispensable pour pouvoir cadencer efficacement les périodes de réveil et les périodes de sommeil, de façon à assurer une communication de bout-en-bout dans le réseau. Dans cette partie, nous décrivons comment cette phase de synchronisation est effectuée et comment des améliorations peuvent réduire sa durée.

### 2.1 Description

MaCARI découpe le temps en cycles globaux qui se répètent périodiquement. Un cycle global commence par une période de synchronisation (cf. 1.1). L'entité qui gère la synchronisation est le CPAN. Le CPAN initie la propagation d'une trame de *beacon* contenant les informations nécessaires pour que tous les nœuds du réseau sachent les intervalles de temps durant lesquels ils doivent rester éveillés et les intervalles de temps pour lesquels ils peuvent se mettre en mode sommeil pour économiser de l'énergie. Ce *beacon* est propagé par la suite en cascade par tous les coordinateurs du réseau d'une manière séquentielle, pour atteindre toutes les entités du réseau<sup>4</sup>.

Pour éviter d'avoir des collisions de *beacons*, le CPAN indique dans le *beacon* l'ordre dans lequel les nœuds doivent diffuser leurs *beacons*. Nous appelons cet ordre l'ordre de synchronisation. Pour illustrer une cascade de *beacon*, nous considérons la topologie de la figure 3.7 qui représente un réseau simple (plus simple que celui présenté sur la figure 3.1) constitué de sept coordinateurs.

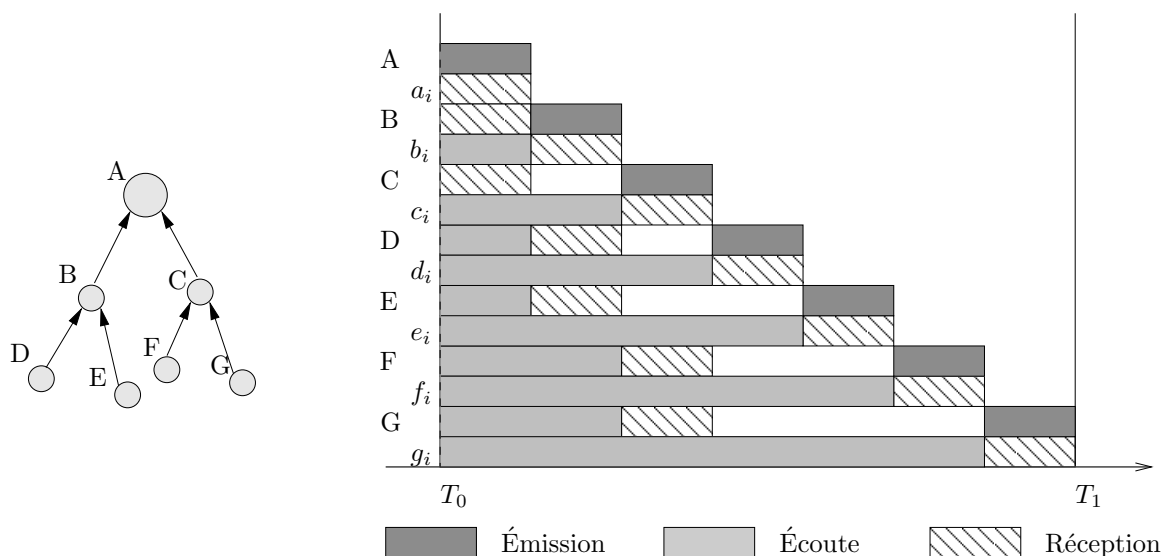


FIG. 3.7 – Un exemple de cascade de *beacons*.

<sup>4</sup>À l'exception des coordinateurs situés à la profondeur maximale; un nœud à la profondeur maximale ne peut pas accepter de nouvelles associations, il n'a donc pas d'entités qui lui sont associées.

Pour simplifier, nous n'avons pas présenté les feuilles sur cette topologie, mais nous considérons qu'à chaque coordinateur est associé un ensemble de feuilles. Sur la figure 3.7,  $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$ ,  $e_i$ ,  $f_i$  et  $g_i$  désignent respectivement l'ensemble des feuilles associées aux coordinateurs A, B, C, D, E, F et G. Nous considérons dans cet exemple que A est le CPAN et qu'il adopte l'ordre de synchronisation en largeur qui est le suivant : (B, C, D, E, F, G)<sup>5</sup>.

A commence la cascade en diffusant le premier *beacon*. Ce *beacon* est reçu par ses coordinateurs fils, B et C, et ses feuilles  $a_i$ . Grâce à l'ordre de synchronisation contenu dans le *beacon*, B sait qu'il est le coordinateur suivant à envoyer le *beacon*. C sait qu'il doit attendre l'envoi de B. B envoie son *beacon* en deuxième, suivi par C. De même, quand D et E reçoivent le *beacon* de B, D sait qu'il doit attendre l'envoi de C, et E sait qu'il doit attendre l'envoi de C et l'envoi de D. Ainsi grâce à l'ordre de synchronisation qui est diffusé dans le *beacon*, les coordinateurs savent à quel instant il faut envoyer le *beacon* qu'ils ont reçu sans générer de collisions. Un fois que tous les coordinateurs ont envoyé leur *beacon*, toutes les entités du réseau (feuilles et coordinateurs) l'auront reçu et seront synchronisées sur  $T_1$ .

## 2.2 Format d'un *beacon* MaCARI

La trame du *beacon* de synchronisation contient des informations qui permettent aux entités du réseau de connaître le découpage temporel du cycle global de MaCARI. Ces informations sont codifiées dans la *payload* MAC de la trame de *beacon*, tel qu'indiqué dans le tableau 3.2.

Octets : 2	1	2	2	variable (118 max)	2
<i>Frame control</i>	<i>Sequence number</i>	<i>PAN ID</i>	<i>Src. address</i>	<i>Beacon payload</i>	FCS
MaCARI header				MaCARI payload	MaCARI footer

TAB. 3.2 – Format du *beacon* MaCARI.

### 2.2.1 L'en-tête du *beacon*

- ***Frame control* (champ de contrôle de la trame)** : ce champ spécifie le type de la trame, d'autres informations concernant le coloriage de SERENA et le découpage temporel de MaCARI.
- ***Sequence number*** : un numéro de séquence de niveau MAC de la trame de *beacon* (modulo 256, géré par la CPAN).
- ***PAN ID*** : l'identifiant du PAN auquel appartient le coordinateur.

---

<sup>5</sup>Nous supposons aussi que les coordinateurs D, E, F et G ne sont pas à la profondeur maximale.

- **Src. address** : adresse courte du coordinateur.

### 2.2.2 Le champ *payload* du *beacon*

La couche physique adoptée pour la pile OCARI est celle de la norme IEEE 802.15.4. La taille maximale de la *payload* physique (contenu véhiculé par la trame physique) est de 127 octets. À cause de cette limitation de taille, la longueur maximale d'un *beacon* est trop petite pour pouvoir y mettre un ordre de synchronisation de taille raisonnable et une durée de période d'activité propre à chaque étoile. Nous avons donc choisi de n'avoir que deux valeurs possibles pour la durée d'une étoile.

Octets : 1	1	4	1	variable (102 max)	variable (7 max)	0/1	0/1
<i>Number of Coordinators</i>	<i>Sequence Number</i>	<i>Global cycle</i>	<i>Activity period duration</i>	<i>Synchronization order</i>	<i>Activity bitmap</i>	<i>Color sequence</i>	<i>Colored slot duration</i>

TAB. 3.3 – Le *payload* du *beacon*.

- **Number of Coordinators** : le nombre de coordinateurs présents dans le réseau et participant à la propagation du *beacon* (le CPAN exclus). Cela correspond au nombre d'adresses courtes dans l'ordre de synchronisation.
- **Sequence Number** : le numéro de séquence du cycle global (modulo 256, géré par le CPAN, utilisé comme base pour l'estampillage temporel).
- **Global cycle** : la durée qui sépare  $T_1$  et le prochain  $T_0$ , exprimée en nombre de périodes de *backoff* (320  $\mu$ s).
- **Activity period duration** : la durée de la période d'activité de chaque étoile dans  $[T_1; T_2]$  (unité 320  $\mu$ s).
- **Synchronization order** : la liste des adresses courtes des coordinateurs du réseau qui participent à la propagation du *beacon*. L'ordre des adresses courtes spécifie l'ordre dans lequel le *beacon* sera propagé dans  $[T_0; T_1]$ .
- **Activity bitmap** : Ce *bitmap* nous permet d'allouer plus de temps pour certaines étoiles. Le nombre de bits significatifs présents dans ce *bitmap* correspond au nombre de coordinateurs présents dans la *short address list*. Un bit à 1 dans la case  $i$  du *bitmap* signifie que la  $i^{eme}$  étoile, dont le coordinateur a son adresse courte dans la case  $i$  de *Synchronization order*, possède  $n$  fois la durée par défaut ( $n$  est un entier commun à toutes les stations).
- **Color sequence** : un entier indiquant le nombre de couleurs utilisées, les couleurs étant allouées de manière consécutive en partant de 0.



- **Colored slot duration** : la durée d'un slot coloré dans  $[T_2; T_3]$ .

Pour construire la *payload* du *beacon*, le CPAN a besoin de connaître les adresses courtes des coordinateurs associés à son réseau. Pour ce faire, à chaque association, une primitive est remontée jusqu'au CPAN pour l'informer de l'adresse courte du coordinateur qui vient de s'associer au réseau. Ayant reçu les adresses courtes de tous les coordinateurs du réseau par ordre d'association, le CPAN est capable de conserver un ordre topologique pour l'ordre de synchronisation qui garantit qu'un coordinateur n'envoie pas son *beacon* avant que son père ne l'ait fait. Cet ordre peut être l'ordre d'association des coordinateurs ou des ordres plus complexes (cf. 2.7.2).

### 2.2.3 Unicité de la *payload* du *beacon*

La *payload* du *beacon* est la même pour tous les *beacons* d'une cascade. En effet, les coordinateurs reçoivent le contenu du *beacon* et le retransmettent sans le modifier. Cela permet d'avoir une certaine flexibilité dans la synchronisation en cas de perte du lien entre un père et un fils, puisqu'il suffit de recevoir un *beacon* transmit de n'importe quel coordinateur du réseau pour avoir les informations nécessaires pour le découpage temporel de MaCARI.

## 2.3 Périodicité de la synchronisation

Le but de la synchronisation est d'informer toutes les entités du réseau du découpage temporel du cycle global. Si ce découpage reste le même pour  $n$  cycles globaux, la synchronisation peut n'être faite qu'une fois tous les  $n$  cycles globaux. Toutefois, il faut s'assurer que les stations ne se désynchronisent pas lorsque la fréquence de synchronisation est réduite.

Une évaluation a été faite pour quantifier la durée au delà de laquelle la dérive des horloges locales des entités cause un décalage conséquent sur le découpage temporel. En supposant que la dérive maximale des composants est de 40 ns par seconde (ce qui est un temps généralement acceptable), la dérive relative entre deux entités quelconques peut atteindre 80 ns. Comme le temps est découpé en périodes de 320  $\mu$ s, le décalage des horloges devient critique quand la périodicité de la synchronisation dépasse 4 s. Dans ce cas, deux intervalles de 320  $\mu$ s, normalement disjoints, peuvent se chevaucher. Nous avons envisagé d'intégrer des intervalles de garde pour contrer ce problème.

## 2.4 Calcul de $T_1$ et de l'instant d'envoi du *beacon*

Le but de la période de synchronisation est de synchroniser l'activité des nœuds du réseau d'une manière relative par rapport à l'instant  $T_1$ .  $T_1$  est calculé grâce à l'ordre de synchronisation envoyé dans la *payload* du *beacon*. Le calcul de  $T_1$  diffère entre coordinateurs et feuilles.

**Dans le cas d'un coordinateur (hormis le CPAN) :** le coordinateur extrait du *beacon* l'ordre de synchronisation et retrouve l'adresse source de la trame de *beacon* dans la liste des adresses courtes de l'ordre de synchronisation<sup>6</sup>. Ayant connaissance de la durée d'envoi d'un *beacon*, le coordinateur est capable de calculer la durée qui sépare l'instant de réception du *beacon* et l'instant auquel il doit envoyer son *beacon*. Pour calculer  $T_1$ , il suffit de récupérer le nombre total de coordinateurs indiqué dans le champ *Number of Coordinators* du *beacon* et de calculer le nombre de coordinateurs devant envoyer le *beacon* à partir de la position de la source. Ainsi, le coordinateur calcule d'une manière relative l'instant  $T_1$ , qui sert comme référence pour le découpage temporel de MaCARI. L'algorithme 1 présente le calcul de  $T_1$  et de l'instant d'envoi de *beacon*. Les deux paramètres essentiels pour le calcul fait par un coordinateur sont donc l'adresse de la station qui a diffusé le *beacon* et la liste des stations qu'il transporte.

---

**Algorithm 1** Algorithme de calcul de  $T_1$  et de l'instant d'envoi du *beacon*.

---

**Pré-requis :** *synchroOrder* = ordre de synchronisation, *sendDuration* = durée d'envoi d'un *beacon*

```

sourcePosition ← -1 (Pour le CPAN)
pour chaque adresse courte de synchroOrder faire
    si adresse courte = adresse source du beacon alors
        sourcePosition ← indice actuel dans la synchroOrder
    fin si
    si le nœud est un coordinateur alors
        si adresse courte = adresse courte du nœud alors
            myPosition ← indice actuel dans la synchroOrder
        fin si
    sinon
        si adresse courte = adresse courte du coordinateur père alors
            myPosition ← indice actuel dans la synchroOrder
        fin si
    fin si
fin pour
si le nœud est un coordinateur alors
    sendTime ← temps actuel + (myPosition - sourcePosition)*sendDuration
fin si
 $T_1$  ← temps actuel + (nombre de coordinateurs - sourcePosition)*sendDuration

```

---

**Dans le cas d'une feuille :** une feuille effectue uniquement le calcul de  $T_1$  en se basant sur l'adresse courte de son père.

## 2.5 L'estampillage des trames

Le délai de traversée de bout-en-bout dans le réseau est une donnée fondamentale pour certaines applications. Nous avons adjoint à MaCARI un mécanisme d'estampillage temporel. Ce mécanisme est un moyen de savoir si une trame a séjourné dans la file d'attente d'une ou de plusieurs entités relais intermédiaires.

Le temps au niveau MaCARI est cadencé à partir d'intervalles de 320  $\mu$ s durant toutes les périodes

---

<sup>6</sup>Le CPAN est le seul coordinateur non présent.

du cycle global. La synchronisation sur l'instant  $T_1$  permet à toutes les entités du réseau de partager une connaissance temporelle commune. Cette connaissance temporelle permet d'estampiller la prise en charge d'une trame de données avec une horloge globale et permet au niveau applicatif de connaître l'instant de création des données (pour les ordonnancer par exemple). Cette information est critique dans la plupart des applications industrielles car elle permet notamment de juger de la fraîcheur des données.

Cette estampille fait partie de l'en-tête application mais est renseignée par MaCARI. La couche application envoie une requête à la couche MaCARI pour récupérer le temps niveau MAC. MaCARI renvoie les deux indications temporelles : le numéro de séquence du cycle global et le nombre de slots de  $320 \mu s$  écoulés depuis le dernier  $T_1$ . En négligeant le temps de traversée des couches entre la couche applicative et la couche MaCARI, cet instant peut être assimilé à l'instant de la soumission des données avec une imprécision liée au temps nécessaire pour convertir les données analogiques provenant du capteur en données numériques prêtes à l'encapsulation réseau (ce temps de soumission peut être assimilé au temps de création des données quand  $[T_3; T_0]$  est nul).

À chaque remontée d'une indication de réception de données de la couche MaCARI à la couche supérieure, MaCARI indique dans la primitive le temps de réception au niveau MAC de la trame. De cette façon, la couche application connaît une image de l'instant de création des données dans l'en-tête application et l'instant de réception niveau MAC, elle peut ainsi estimer la fraîcheur des données en fonction de la durée écoulée entre ces deux instants.

## 2.6 Économie d'énergie

Durant  $[T_0; T_1]$ , les nœuds économisent de l'énergie en fonction de leur type. Les feuilles peuvent se mettre en mode sommeil dès qu'elles ont reçu leur *beacon*. Les coordinateurs peuvent dormir après avoir envoyé le *beacon* d'une part, et entre l'instant de réception du *beacon* et l'instant de son envoi d'autre part (si la durée entre ces deux instants est suffisamment grande).

Nous considérons qu'une entité consomme en continue la même quantité d'énergie quand elle est en mode éveillé. Nous faisons l'hypothèse simplificatrice en ne considérant pas les différents niveaux de consommation pour les différents états d'une entité (émission, réception, *idle*, commutation d'un état à l'autre).

Si les coordinateurs dorment uniquement après avoir envoyé leur *beacon*, tous les nœuds économisent de l'énergie durant environ 50% de la durée de  $[T_0; T_1]$  en moyenne. Supposons que nous ayons  $n$  coordinateurs et  $m$  feuilles dans le réseau. L'énergie totale consommée par toutes ces entités est  $n^2$ . Il s'agit d'une consommation en prenant comme unité celle nécessaire à l'émission d'un *beacon*.

$$\text{L'énergie totale consommée sera : } \sum_{i=1}^n i = \frac{n*(n+1)}{2} = \frac{n^2+n}{2}.$$

Le ratio de l'énergie totale consommée est donc :

$$\frac{\text{temps actif}}{\text{temps total}} = \frac{\frac{n^2+n}{2}}{n^2} = \frac{n^2}{2*n^2} + \frac{n}{2*n^2} = \frac{1}{2} + \frac{1}{2*n}.$$

Ainsi, la consommation énergétique moyenne totale durant  $[T_0; T_1]$  pour  $n$  coordinateurs et  $m$  feuilles est donnée par la formule suivante :

$E_{T_0T_1} = (m + n) * D_{T_0T_1}/2 + \frac{(m+n)*D_{T_0T_1}}{2*n}$  où  $D_{T_0T_1}$  est la durée de la période de synchronisation.

## 2.7 Réduction de la durée de la période de synchronisation

La période de synchronisation est une étape indispensable mais elle constitue une surcharge temporelle car les informations envoyées durant cette période ne sont pas des données applicatives. Dans cette partie nous décrivons des méthodes pour réduire la durée de la période de synchronisation et diminuer ainsi sa surcharge de la bande passante du réseau.

### 2.7.1 Amélioration par envoi anticipé

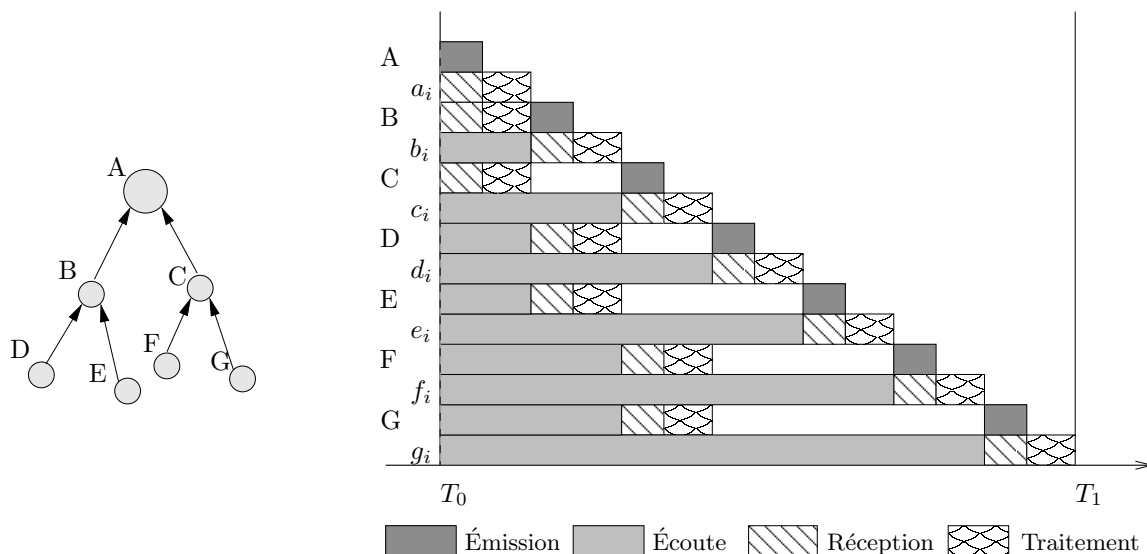
Sur la figure 3.7, nous avons présenté une version simplifiée de ce qui se passe durant la période de synchronisation. En fait, à la réception d'un *beacon*, un temps non négligeable de traitement de la trame de *beacon* doit être pris en compte, en raison de la puissance de calcul limitée offerte par les microcontrôleurs utilisés pour les réseaux de capteurs sans fil. Ainsi, le rectangle qui représente la réception doit être décomposé en deux parties : (i) la partie qui correspond à la durée de réception des bits du *beacon* qui dépend du débit des modules radio et de la longueur de la trame de *beacon*, (ii) et la partie qui correspond à la durée de traitement du *beacon*, qui dépend de la fréquence du microprocesseur et des tâches de calcul à effectuer. Le temps de traitement comprend la désencapsulation et l'analyse de la trame de *beacon*, le calcul de  $T_1$  et de l'instant d'envoi, et la construction du *beacon* à envoyer.

La figure 3.8 représente la même cascade de *beacons* que celle représentée sur la figure 3.7, mais en ajoutant le temps de traitement<sup>7</sup>.

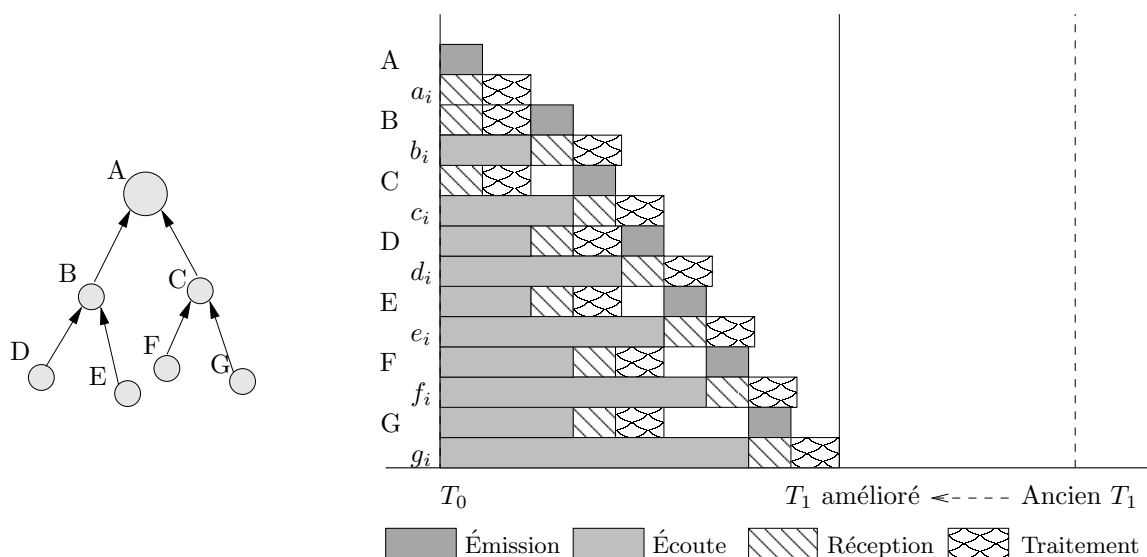
Nous remarquons sur la figure 3.8 que les coordinateurs peuvent anticiper leur instant de transmission de *beacon* sans générer des collisions. Prenons le cas du coordinateur C. Après avoir reçu et traité le *beacon* de A, C attend l'envoi du *beacon* de B avant d'envoyer son propre *beacon*. L'unité de temps que nous considérons pour attendre un envoi de *beacon* englobe la durée de transmission et la durée de traitement. En informant le nœud C du fait qu'il ne faut attendre que le temps de transmission de B, C peut profiter du temps de traitement du *beacon* de B pour anticiper sa transmission. La durée  $[T_0; T_1]$  peut ainsi être réduite.

---

<sup>7</sup>pour simplifier, nous avons représenté sur la figure 3.8 la même durée de traitement pour les feuilles et les coordinateurs, alors que la feuille a moins de traitement que le coordinateur. Cela ne change en rien l'explication.


 FIG. 3.8 – Cascade de *beacons* avec temps de traitement.

La figure 3.9 montre le gain que nous pouvons obtenir si les coordinateurs peuvent anticiper l'envoi de *beacon*.


 FIG. 3.9 – Cascade de *beacons* avec envois anticipés.

**Modification du format du *beacon* pour prendre en compte l'amélioration** Grâce aux adresses courtes, il est possible de déterminer si un coordinateur quelconque est le fils d'un autre coordinateur. L'algorithme 2 permet de calculer la profondeur d'une entité à partir de son adresse courte. Ayant sa profondeur, avec l'algorithme 3 nous pouvons retrouver l'adresse courte du père d'un coordinateur. Notons que tous les coordinateurs du réseau connaissent les trois paramètres  $Rm$ ,  $Lm$  et  $Cm$  qui sont utilisés dans les calculs des algorithmes qui suivent.

L'algorithme 4 décrit comment les envois anticipés sont pris en compte dans le calcul de  $T_1$  et de l'instant d'envoi du *beacon*. Les algorithmes 2 et 3 sont appelés par l'algorithme 4.

---

**Algorithm 2** Algorithme de calcul de la profondeur en fonction de l'adresse courte.

---

**Pré-requis :**  $Lm$  = profondeur maximum de l'arbre,  $Rm$  = nombre de fils coordinateur maximum,  $Cm$  = nombre de fils maximum,  $adr\_courte$ ,  $adr\_courante$ ,  $prof\_courante$

```

si  $adr\_courte = 0$  alors
    return 0
fin si
si  $adr\_courte \geq (adr\_courante + 1 + Cskip(prof\_courante) * Rm)$  et  $adr\_courte \leq (adr\_courante + 1 + Cskip(prof\_courante) * Rm) + (Cm - Rm)$  alors
    return  $prof\_courante + 1$ 
fin si
si  $Rm = 1$  alors
    si  $adr\_courte = adr\_courante + 1$  alors
        return  $prof\_courante + 1$ 
    fin si
    si  $prof\_courante + 1 < Lm$  alors
        return  $prof(adr\_courte, adr\_courante + 1, prof\_courante + 1)$ 
    fin si
sinon
    pour  $i = 0$  to  $Rm - 1$  faire
        si  $adr\_courte = adr\_courante + 1 + Cskip(prof\_courante) * i$  alors
            return  $prof\_courante + 1$ 
        fin si
        si  $adr\_courte > adr\_courante + 1 + Cskip(prof\_courante) * i$  et  $adr\_courte < adr\_courante + 1 + Cskip(prof\_courante) * (i + 1)$  alors
            si  $prof\_courante + 1 < Lm$  alors
                return  $prof(adr\_courte, adr + 1 + Cskip(prof\_courante) * i, prof\_courante + 1)$ 
            fin si
        fin si
    fin pour
fin si

```

---

Notez que sur l'exemple de la figure 3.9, nous avons pu anticiper l'envoi de tous les coordinateurs sauf celui de B, car B n'a pas d'attente à effectuer entre l'instant de réception et l'instant d'envoi. Ce cas se pose à chaque fois qu'un fils doit envoyer son *beacon* juste après son père. Par la suite nous appelons ce cas une attente inutile.

### 2.7.2 Amélioration de l'ordre de synchronisation

L'amélioration par envoi anticipé se base sur un ordre de synchronisation donné. Cet ordre lui-même peut être optimisé de façon à minimiser le nombre de fois où un fils doit envoyer son *beacon* juste après son père.

Sur l'exemple de la figure 3.10, nous avons pu éviter de mettre le fils E juste après son père D mais nous n'avons pas pu éviter que le fils B soit juste après son père A. Par la suite, nous allons expliquer des règles à appliquer pour construire un ordre de synchronisation amélioré en plusieurs étapes.

**Algorithm 3** Algorithme de vérification de lien de parenté.

**Pré-requis :**  $Lm$  = profondeur maximum de l'arbre,  $Rm$  = nombre de fils coordinateur maximum,  $Cm$  = nombre de fils maximum,  $adr\_fils$ ,  $adr\_pere$ ,  $prof(adresse, 0, 0)$  retourne la profondeur d'une entité connaissant son adresse courte  
 profondeur du fils  $\leftarrow prof(adr\_fils, 0, 0)$   
 profondeur du pere  $\leftarrow prof(adr\_pere, 0, 0)$   
**si**  $profondeur\ du\ pere = (profondeur\ du\ fils - 1)$  **alors**  
     **pour**  $i = 0$  to  $Rm$  **faire**  
         **si**  $adr\_pere = adr\_fils - 1 - i * Cskip(profondeur\ du\ fils - 1)$  **alors**  
             **return true**  
         **fin si**  
     **fin pour**  
**sinon**  
     **return false**  
**fin si**  
**return false**

**Algorithm 4** Algorithme du calcul de  $T_1$  et de l'instant d'envoi du *beacon*  $sendTime$  avec anticipation d'envoi.

**Pré-requis :**  $synchroOrder$  = ordre de synchronisation,  $sourcePosition$  = position de l'adresse source du *beacon* dans  $synchroOrder$ ,  $myPosition$  = position de l'adresse du coordinateur du *beacon* dans  $synchroOrder$ ,  $sendDuration$  = durée d'envoi d'un *beacon*,  $processDuration$  = durée de traitement,  $nbProcess$  = le nombre de fois où un fils suit son père dans  $synchroOrder$ ,  $nbProcessSend$  = le nombre de fois où un fils suit son père dans  $synchroOrder$  uniquement entre  $sourcePosition$  et la  $myPosition$ ,  $parentOf(adr1, adr2)$  retourne vrai si  $adr1$  est l'adresse du père de l'entité ayant  $adr2$  comme adresse courte  
**pour** chaque adresse courte dans le  $synchroOrder$  **faire**  
     **si** deuxième adresse ou plus, et  $parentOf(adresse\ courte\ actuelle, adresse\ courte\ précédente)$  **alors**  
          $nbProcess \leftarrow nbProcess + 1$   
     **fin si**  
     **si** adresse courte = adresse courte de la source du *beacon* **alors**  
          $sourcePosition \leftarrow indice\ actuel$   
          $nbProcess \leftarrow 0$   
     **fin si**  
     **si** adresse courte = adresse courte du nœud **alors**  
          $myPosition \leftarrow indice\ actuel$   
          $nbProcessSend \leftarrow nbProcess$   
     **fin si**  
**fin pour**  
 $sendTime = temps\ actuel + (myPosition - sourcePosition) * sendDuration + nbProcessSend * processDuration$   
 $T_1 = temps\ actuel + (nombre\ de\ coordinateurs - sourcePosition) * sendDuration + nbProcess * processDuration$

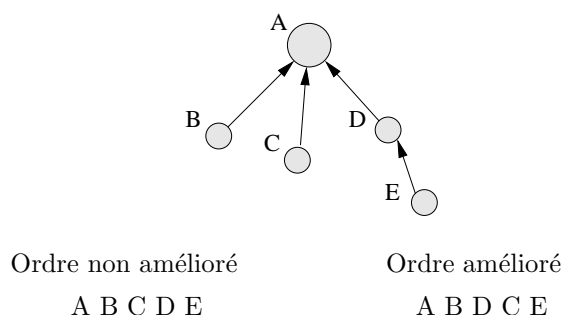


FIG. 3.10 – Exemple d'ordres de synchronisation. Un autre ordre amélioré aurait pu être : A D C B E.

**Parcours en largeur** Un parcours en largeur visite tous les nœuds d'une profondeur avant de passer aux nœuds de la profondeur suivante, et en commençant par la racine. L'exemple de l'ordre de synchronisation non amélioré de la figure 3.10 est un exemple de parcours en largeur. Dans cet exemple, le parcours en largeur nous génère deux cas d'attente inutile.

**Théorème 1.**  $AI(ord\_larg(top)) \leq prof(top)$ . Le nombre d'attente inutiles peut être majoré par la profondeur de l'arbre.

**Preuve 1.** Un fils suit son père dans l'ordre  $ord\_prof(top)$  si et seulement si le fils est le premier de sa profondeur et si le père est le dernier de sa profondeur dans l'ordre  $ord\_larg(top)$ . Ce cas se présente, au plus, une fois par profondeur. Ainsi,  $AI(ord\_larg(top)) \leq prof(top)$ .

En adoptant un parcours en largeur, le nombre d'attentes inutiles est majoré par la profondeur de l'arbre. Soient  $AI(ord(top))$  le nombre d'attentes inutiles dans l'ordre  $ord$  appliqué sur une topologie donnée  $top$ ,  $ord\_larg(top)$  le parcours en largeur appliqué à la topologie  $top$  et  $prof(top)$  la profondeur de la topologie  $top$ .

**Permutation** D'après le théorème 1, un cas d'attente inutile n'arrive au plus qu'une fois par profondeur dans un parcours en largeur. Ainsi, nous pouvons attribuer un cas d'attente inutile à la profondeur au lieu de l'attribuer à un nœud. Ce cas d'attente inutile peut être évité en appliquant une permutation selon la règle suivante : si le nombre de nœuds de la profondeur est au moins 3, ou si le nombre de nœuds de la profondeur est 2 et les deux nœuds ont le même père, nous permutons le dernier nœud avec son précédent.

D'autres règles de permutation et de déplacement de nœuds dans l'ordre de synchronisation pour éviter les cas d'attentes inutiles sont étudiées dans un rapport interne à notre équipe de recherche [55].



### 3 Segmentation

L'utilité de la période de synchronisation ne se limite pas à l'économie d'énergie. En utilisant le *beacon* comme un moyen de diffusion d'informations, nous pouvons mettre en place un mécanisme de découpage temporel qui améliore l'utilisation du canal et qui garantit l'accès au médium pour un trafic prioritaire.

Dans cette partie, nous détaillons comment MaCARI réalise une segmentation temporelle à deux niveaux : (i) une segmentation pour réserver le canal successivement pour chaque étoile, et ainsi permettre à chaque coordinateur de gérer l'activité de son étoile en éliminant les interférences causées par l'activité d'autres étoiles du réseau, (ii) et une segmentation pour garantir le relais entre les coordinateurs à un type de trafic prioritaire et contraint temporellement.

#### 3.1 Description de la segmentation

Nous décrivons dans ce qui suit la segmentation temporelle en périodes d'activité, période de routage et intervalles de relais garanti.

##### 3.1.1 Séquencement des activités des étoiles

Se synchroniser par rapport à  $T_1$  permet à tous les nœuds du réseau de connaître les périodes d'activité et les périodes de sommeil. Pour éviter d'avoir une forte contention durant laquelle toutes les entités essaient d'accéder au médium durant une période commune, nous proposons une segmentation dans la période d'activité basée sur le regroupement des entités en étoile, en réservant à priori, le canal à chaque étoile d'une manière séquentielle.

Grâce à cette segmentation, chaque coordinateur gère l'activité de son étoile durant la période d'activité qui lui est allouée, indépendamment du reste de l'activité du réseau. Ainsi, le coordinateur est capable d'allouer des slots de temps à accès garanti pour une feuille dans son étoile en étant sûr que ce slot ne sera pas utilisé par d'autres coordinateurs du réseau.

Les échanges durant la période d'activité entre les coordinateurs et leurs feuilles sont gérés selon l'algorithme CSMA/CA slotté pour le trafic non-prioritaire et en GTS pour le trafic prioritaire GTS comme le prévoit la norme IEEE 802.15.4. Grâce à ce séquencement, les GTS alloués dans une étoile ne risquent pas d'être en conflit avec d'autres GTS alloués dans d'autres étoiles du réseau. Le trafic prioritaire est ensuite relayé entre les coordinateurs père-fils durant les intervalles de relais. Le trafic non-prioritaire est routé durant l'intervalle  $[T_2; T_3]$  en utilisant la méthode d'accès CSMA/CA slotté.

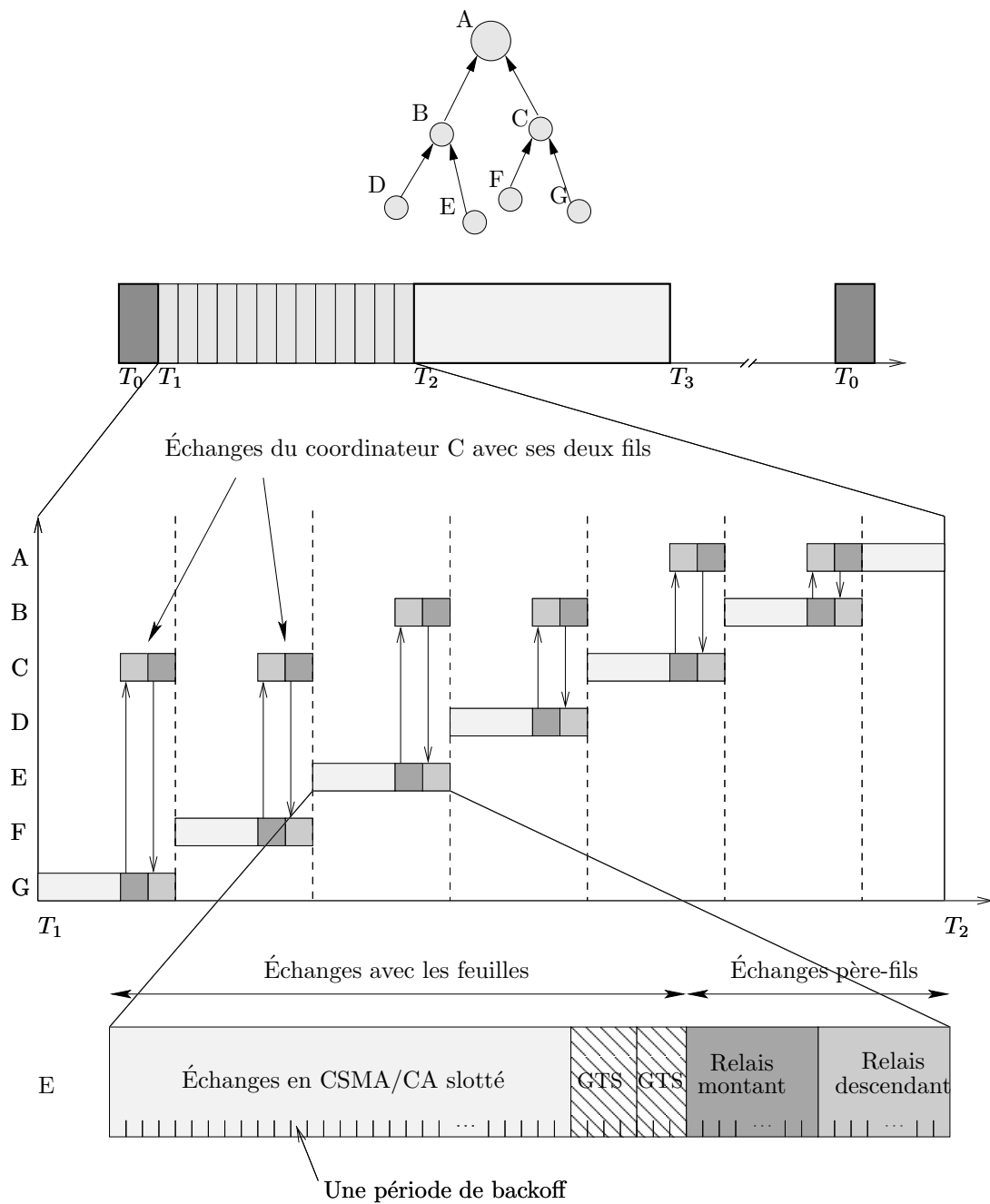


FIG. 3.11 – Séquencement des périodes d'activité des étoiles.

### 3.1.2 Période de routage

Pour assurer une connectivité entre les coordinateurs et leur permettre de communiquer entre eux, une période de routage suit la période des activités des étoiles. Durant cette période, les coordinateurs acheminent les données récoltées durant les différentes périodes d'activités des étoiles (ou communiquent des informations aux actionneurs).

Durant cette période, ce sont uniquement les coordinateurs qui sont actifs et communiquent entre eux en appliquant le CSMA/CA slotté.

### 3.1.3 Relais garanti

Pour assurer une qualité de service pour le trafic échangé entre les coordinateurs, MaCARI réserve des intervalles de temps pour chaque couple père-fils. Ces intervalles de temps suivent la fin de la période d'activité du coordinateur fils. On appelle ces intervalles de temps les intervalles de relais garanti.

De cette façon, des slots de temps sont réservés pour faire passer un trafic prioritaire entre les coordinateurs. Comme le protocole CSMA/CA slotté est un protocole aléatoire qui ne garanti ni l'accès au médium, ni des transmissions sans collision, les intervalles de relais garanti permettent de réserver l'accès au médium pour un trafic prioritaire.

Le couple de coordinateurs père-fils est composé des deux seules entités actives dans le réseau durant l'intervalle de relais et l'accès au médium est donc à négocier entre ces deux entités. Pour éviter d'avoir des collisions entre les trames envoyées par le coordinateur fils et celles envoyées par le coordinateur père, un protocole d'échanges entre ces deux coordinateurs est défini de la manière suivant : en début de l'intervalle de relais, le coordinateur fils initie l'échange en envoyant les trames à relayer à son coordinateur père. Les trames sont acquittées une par une. Quand le fils finit d'envoyer ses trames, il envoie une trame de contrôle indiquant la fin du relais montant. Cela permet au coordinateur père de commencer le relais descendant et de faire passer les trames de type prioritaire en attente à destination du coordinateur fils.

Dans le cas où le coordinateur père n'a pas (ou plus) de trames à relayer vers son fils, ce qui peut être souvent le cas, il l'indique au coordinateur fils au moyen d'un bit de l'acquittement. Le fils est alors informé qu'il peut utiliser le temps prévu pour le relais descendant.

## 3.2 Économie d'énergie

Avec ce découpage temporel, les entités du réseau économisent de l'énergie. Soit  $D_{T_1T_2}$  la durée de la période  $[T_1; T_2]$  et  $D_{T_2T_3}$  la durée de la période  $[T_2; T_3]$ . Nous considérons que l'intervalle de relais garanti dure  $1/3$  de la durée d'une période d'activité d'une étoile. Le trafic prioritaire est orienté par l'arbre et cela génère des engorgements auprès de la racine (qui est souvent le destinataire de ce flux). Une gestion

des allocations des intervalles de temps est nécessaire pour réserver des ressources nécessaire en fonction de la position du nœud dans l'arbre. Dans un premier temps, nous avons choisi une solution statique qui consiste à utiliser le tiers de la période d'activité pour faire passer ce flux. Il est intéressant de prévoir une réservation dynamique en fonction du nombre de descendants et de la taille du trafic. Ce point sera discuté dans 3.4.1.

Un coordinateur est actif dans  $[T_1; T_3]$  durant  $D_{T_1T_2}/n + (D_{T_1T_2}/n) * 1/3 * Rm + D_{T_2T_3}$  et une feuille sera active durant  $(D_{T_1T_2}/n) * 2/3$ . Où  $n$  est le nombre de coordinateurs dans le réseau et  $Rm$  est le nombre maximum de fils coordinateurs par coordinateur.

Avec les hypothèses de consommation de 2.6 et en considérant que l'arbre est complet en terme de nombre de coordinateurs (tous les coordinateurs ont  $Rm$  fils coordinateurs), la consommation énergétique peut être assimilée à la durée d'activation moyenne durant  $[T_1; T_3]$ , avec  $n$  coordinateurs et  $m$  feuilles, est donnée par la formule suivante :

$$E_{T_1T_3} = n * (D_{T_1T_2}/n + (D_{T_1T_2}/n) * 1/3 * Rm + D_{T_2T_3}) + m * ((D_{T_1T_2}/n) * 2/3) = D_{T_1T_2} + D_{T_1T_2} * 1/3 * Rm + n * D_{T_2T_3} + m/n * (D_{T_1T_2}) * 2/3 = (1 + Rm/3 + 2m/3n) * D_{T_1T_2} + n * D_{T_2T_3}.$$

Avec la formule calculée dans 2.6, la consommation énergétique moyenne totale du réseau dans  $[T_0; T_3]$  est :

$$E_{T_0T_3} = E_{T_0T_1} + E_{T_1T_3} = (m+n) * D_{T_0T_1}/2 + \frac{(m+n) * D_{T_0T_1}}{2 * n} + (1 + Rm/3 + 2m/3n) * D_{T_1T_2} + n * D_{T_2T_3} = (1/2 + 1/(2 * n)) * (m+n) * D_{T_0T_1} + (1 + Rm/3 + 2m/3n) * D_{T_1T_2} + n * D_{T_2T_3}.$$

### 3.3 Calcul des instants des périodes d'activité

La segmentation se base sur l'ordre de synchronisation et sur les informations de la *payload* du *beacon*. Pour ce faire, nous avons considéré, en premier temps comme une solution simpliste, que l'ordre de séquençement des périodes d'activité de chaque étoile correspond à l'inverse de l'ordre de synchronisation, chaque étoile étant représentée par l'adresse courte de son coordinateur. L'algorithme 5 détaille le calcul des périodes d'activité.

### 3.4 Délai garanti

Grâce aux intervalles de relais garanti, MaCARI garantit un délai de bout-en-bout sous réserve d'un dimensionnement convenable des intervalles de relais garanti en fonction de la quantité de trafic prioritaire à faire passer.

À partir d'un ordre de synchronisation, il est possible de déduire très facilement (au moins) deux ordres de séquençement des périodes d'activité dans  $[T_1; T_2]$  : soit l'ordre lui-même, soit l'ordre inverse. En adoptant l'ordre de synchronisation pour les périodes d'activité, nous favorisons le relais d'un trafic

---

**Algorithm 5** Algorithme de calcul des périodes d'activité.

**Pré-requis :** nbCoord = nombre de coordinateurs, nbFils = nombre de fils coordinateurs, ActivityDuration = durée d'une période d'activité, sourceAdr = adresse source du coordinateur ou de la feuille, synchroOrder = ordre de synchronisation, childrenList = liste de structures contenant les adresses courtes et le début d'activité des fils coordinateurs, ActivityBitmap = bitmap indiquant si le coordinateur à la  $i$ -ème position du synchroOrder a une période d'activité double ou non  
tempsEcoule  $\leftarrow$  0

```
pour chaque adresse courte du synchroOrder faire
  si le nœud est un coordinateur alors
    si adresse courte du synchroOrder = l'adresse courte du coordinateur alors
      début de la période d'activité  $\leftarrow T_1 + \text{tempsEcoule}$ 
    sinon
      pour tous les fils coordinateurs du nœud faire
        si adresse courte de la synchroOrder = l'adresse courte d'un fils alors
          début de la période d'activité de ce fils  $\leftarrow T_1 + \text{tempsEcoule}$ 
        fin si
      fin pour
    fin si
  sinon
    si adresse courte de la synchroOrder = l'adresse courte du coordinateur père de la feuille alors
      début de la période d'activité  $\leftarrow T_1 + \text{tempsEcoule}$ 
    fin si
  fin si
si le bitmap est vrai pour cet indice alors
   $\text{tempsEcoule} \leftarrow \text{tempsEcoule} + 2 * \text{ActivityDuration}$ 
sinon
   $\text{tempsEcoule} \leftarrow \text{tempsEcoule} + \text{ActivityDuration}$ 
fin si
fin pour
```

---

descendant. En adoptant l'ordre inverse nous favorisons le relais d'un trafic montant.

Prenons l'exemple présenté sur la figure 3.11. Supposons que le coordinateur G ait reçu d'une de ses feuilles, une trame de type prioritaire à destination du CPAN A. Cette trame est relayée de G vers C, puis de C vers A en un seul cycle (si la taille des intervalles de relais le permet). Si A veut envoyer une trame de type prioritaire à G, A va recevoir la trame d'une de ses feuilles dans un cycle global, la relayer à C dans le cycle global suivant, et enfin, C va la relayer à G dans le troisième cycle global.

En adoptant un seul sens de séquençement des périodes d'activité, les trames qui sont à contre sens sont pénalisées et subissent un délai de bout-en-bout qui est fonction de la profondeur maximale de la topologie, et de la position relative de la source par rapport à celle de la destination dans l'arbre. Ce délai peut atteindre au pire des cas  $(Lm + 1) * \text{dureeDuCycleGlobal}$ .

Pour des applications où le trafic n'est pas majoritairement dans un seul sens, MaCARI peut alterner le séquençement des périodes d'activité à chaque cycle global. De cette manière, le délai de bout-en-bout est majoré par  $3 * \text{dureeDuCycleGlobal}$  au pire des cas, comme nous allons le voir dans la suite.

Prenons l'exemple illustré sur la figure 3.12 où le nœud  $X$ , situé à la profondeur  $Lm$ , veut envoyer une trame de type prioritaire au nœud  $Y$ , situé à la profondeur  $Lm$  de sorte que le parent commun des deux nœuds soit le CPAN. La longueur du chemin à parcourir par la trame est  $2 * Lm$  liens. Le pire des cas se pose quand lors du premier relais, le séquençement est en sens descendant. Dans ce cas, la trame ne parcourt qu'un saut de son chemin. Le séquençement du cycle suivant est dans le sens montant. La trame arrive en fin de cycle jusqu'au CPAN. Le cycle suivant est d'un sens descendant et la trame parcourt le reste du chemin jusqu'à  $Y$  (dans l'hypothèse où le réseau n'est pas saturé pour ce type de trafic).

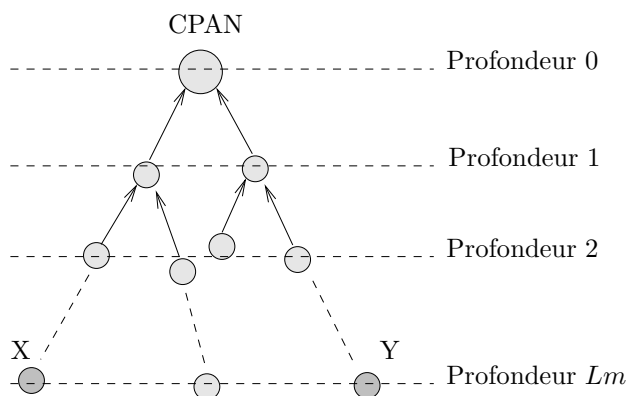


FIG. 3.12 – Un exemple de topologie avec les différentes profondeurs.

### 3.4.1 Adaptation de la taille des intervalles de relais

Les applications des réseaux de capteurs reposent souvent sur la présence d'un puits de données. Ce puits est typiquement la racine de l'arbre. Cela cause une forte congestion des liens utilisés durant les intervalles de relais garanti, notamment pour ceux qui sont proches de la racine.

Pour cela, les intervalles de temps alloués pour le relais garanti doivent être dimensionnés en prenant en compte la charge offerte au réseau de type prioritaire. Nous proposons dans la suite deux méthodes complémentaires pour améliorer le dimensionnement de la taille des intervalles de relais : une distribuée et appliquée par chaque coordinateur en local, et une autre centralisée et effectuée par le CPAN.

**Calcul en local selon le trafic cumulé** La durée allouée à chaque coordinateur est découpée en trois parties, comme le montre la figure 3.11, (i) une période d'accès en CSMA/CA slotté pour les échanges entre le coordinateur et ses feuilles, (ii) des slots GTS alloués aux feuilles qui génèrent ou consomment du trafic prioritaire, et (iii) les intervalles de relais pour communiquer avec le coordinateur père.

Jusqu'à présent, pour des raisons de simplicité, dans cette version de MaCARI nous avons considéré que les intervalles de relais constituent le tiers de la période d'activité. De cette façon, un coordinateur père se réveille uniquement durant le dernier tiers de la période d'activité de son fils pour économiser de l'énergie.

Sans cette contrainte, un coordinateur peut décider de communiquer avec son père au moment qu'il juge nécessaire pour pouvoir faire passer le trafic prioritaire avant la fin de sa période d'activité.

Pour éviter que les feuilles n'interfèrent avec la communication père-fils, les feuilles cessent d'envoyer des messages à la réception d'une trame à destination du coordinateur père. Pour cela, l'adresse du coordinateur père est communiquée aux feuilles au moment de leur association.

**Calcul par le CPAN en fonction des indications de trafic et du nombre de descendants** Le CPAN est capable d'allouer à certains nœuds des périodes d'activité de tailles plus grandes. Pour une question d'optimisation de la taille de trame de *beacon*, nous avons choisi de mettre en place deux tailles différentes uniquement : (i) une taille qui correspond au nombre maximum de feuilles par coordinateur et (ii) une taille qui fait le double de la taille précédente. L'intérêt de doubler la taille est d'augmenter l'intervalle de relais et non pas d'augmenter la période d'activité intra-étoile.

Le CPAN connaît le nombre de coordinateurs descendants de chaque coordinateur, et il reçoit les notifications des allocations de GTS remontées par chaque coordinateur. Avec ces deux informations, le CPAN est capable d'estimer la charge de trafic que chaque coordinateur subit, et ainsi, d'allouer une taille double pour certains coordinateurs.

### 3.5 Schéma récapitulatif

La figure 3.13 montre comment les trames sont classées dans des files d'attente selon leur type et leur destination. Cette représentation correspond à la gestion des files d'attente d'un coordinateur. À la réception d'une trame de la couche réseau, la couche MaCARI vérifie le type de la trame. Si la trame est de type non-prioritaire et non destinée à une feuille, elle est mise dans la file d'attente appelée Routage dans la figure, dans laquelle sont mises toutes les trames à échanger durant l'intervalle  $[T_2; T_3]$ , ceci correspond au test numéro 1 sur la figure 3.13. Si la trame est de type prioritaire ou à destination d'une feuille qui appartient au coordinateur, elle est mise dans une file d'attente en fonction du prochain saut. Si le prochain est une feuille qui appartient au coordinateur, la trame est mise dans la file d'attente intra-étoile. Si la trame est à destination du père ou d'un de ses fils, la trame est mise dans la file d'attente qui correspond à la destination. Ceci correspond au test numéro 2 de la figure 3.13.

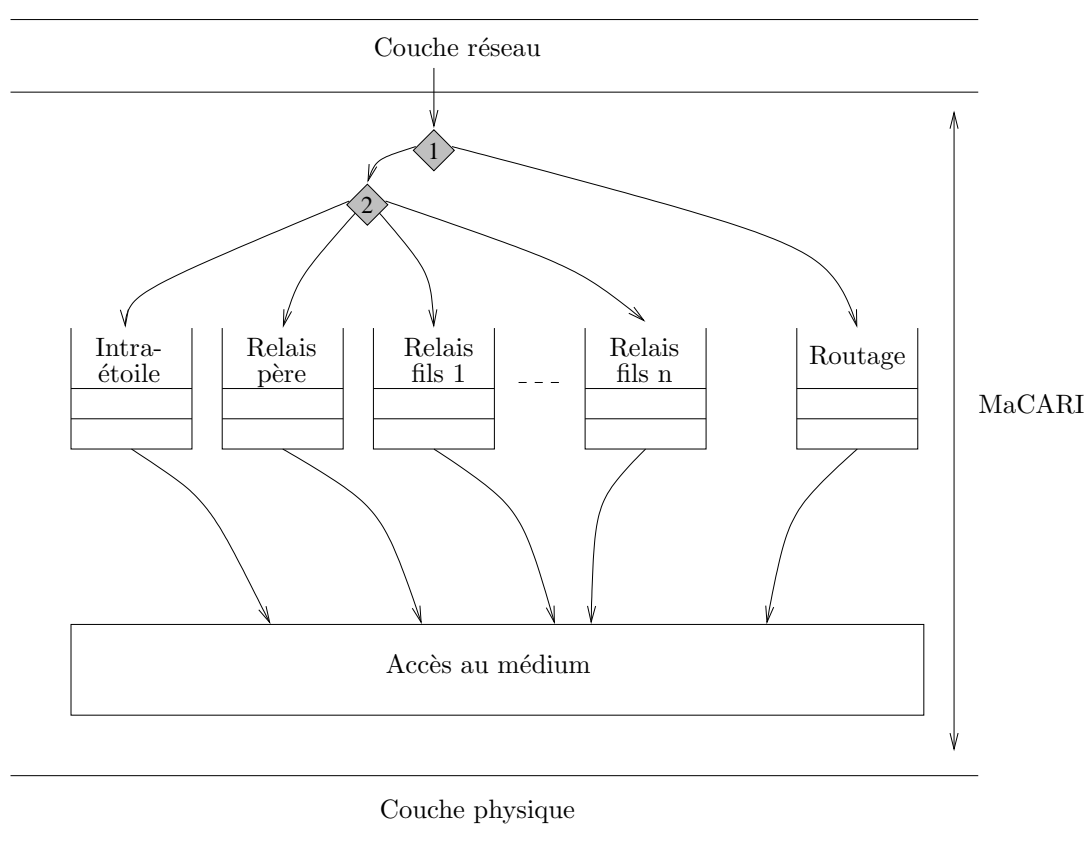


FIG. 3.13 – Gestion des files d'attente selon le type de trafic et la destination.

Ces files d'attente sont sollicitées selon les intervalles de temps de MaCARI. Quand le coordinateur est dans l'intervalle de la période d'activité de son étoile, il sollicite la file intra-étoile, il est de même pour les autres files. Ainsi, le trafic à échanger dans chacun des intervalles de temps est classé dans une file d'attente différente. Il sera intéressant dans la suite d'étudier des priorités différentes à l'intérieur



même de chacune des files d'attente.

Pour le cas d'une feuille, il n'existe qu'une seule file d'attente qui est la file intra-étoile puisque les feuilles ne transmettent que durant la période d'activité de leur étoile.

# Chapitre 4

## Résultats

Dans ce chapitre, nous expliquons la démarche que nous avons adoptée pour valider notre protocole MaCARI. Cette démarche englobe de la simulation, du maquetage et de l'étude analytique.

Dans la première partie, nous commençons par détailler comment nous avons moduler les différents aspects de MaCARI sur le logiciel de simulation, ensuite nous décrivons nos outils de maquetage et le matériel utilisé. Nous présentons dans la deuxième partie les résultats que nous avons obtenus concernant la période de synchronisation. Dans la troisième partie, nous étudions les résultats obtenus en terme de performance du réseau sur la partie segmentation. Nous terminons dans la quatrième partie par une étude analytique pour quantifier le gain énergétique et le délai de bout-en-bout selon différentes configurations.

### 1 Validation

Une démarche de simulation et de maquetage a été nécessaire pour valider les différents aspects de notre proposition. Nous expliquons dans ce chapitre comment nous avons moduler et implémenter le protocole MaCARI, en insistant sur les aspects sans fil de la simulation <sup>1</sup>. En suite, nous décrivons brièvement comment nous avons procéder pour maquetter notre solution.

#### 1.1 Simulation

La simulation est le fait de reproduire le comportement dans le temps d'un phénomène réel en utilisant un modèle qui dans notre cas sera développé et activé à partir d'outils logiciels spécifiques. Cela permet d'observer l'évolution d'un système en faisant varier certains paramètres qui affectent son comportement. La simulation permet de pousser plus loin des tests qui ne sont pas toujours réalisables à cause de contraintes de ressources (temps, coût du matériel qu'on souhaite tester et/ou de la main

---

<sup>1</sup>Le travail de modulation et de simulation était le sujet d'une publication dans un journal international [56].

d'œuvre pour le déploiement d'une solution réelle... etc).

NS2 (*Network Simulator 2*) est un logiciel libre de simulation par événements discrets largement utilisé dans la recherche académique et dans l'industrie. Il met à disposition des modèles de protocoles de réseaux sans fil et filaires comme l'UDP, le TCP/IP, la couche MAC de 802.11, la couche MAC de 802.15.4... etc. Grâce à cet outil de l'univers du logiciel libre, il est possible de lui ajouter de modèles correspondant à des protocoles de tous les niveaux du modèle OSI (*Open Systems Interconnection*) .

Il est basé sur l'utilisation de langages de scripts pour la commande des simulations (tcl/tk) alors que le cœur des simulations est implémenté avec le langage C++. Les modèles de protocoles sont donc implémentés en C++ et le scénario de simulation est spécifié en tcl par l'utilisateur qui précise à ce niveau les protocoles qu'il souhaite utiliser.

NS2 est fourni avec une interface graphique, nommée NAM (*Network AniMator*) , permettant d'illustrer le fonctionnement des réseaux, car il permet la visualisation de l'activité du réseau (les échanges de trames, le positionnement des nœuds, le déplacement des nœuds...).

### 1.1.1 Simulation du médium sans fil

La nature du médium sans fil fait que l'émission d'un signal peut être détectée par toute antenne suffisamment proche qui est en mode réception. On parle de portée pour désigner la distance jusqu'à laquelle un signal émis par une antenne reste compréhensible.

En se propageant dans l'air, le signal est soumis à une atténuation de puissance qui augmente en fonction de la distance et qui finit par rendre le signal incompréhensible. Dans cette partie, nous allons décrire les modèles de propagation les plus cités qui modélisent l'atténuation du signal radio utilisé dans l'univers des WLAN (*Wireless Local Area Network*) et WPAN (*Wireless Personal Area Network*).

La distance n'est pas le seul facteur qui cause une atténuation de puissance : la propagation d'un signal radio dans l'air est altérée par la présence d'objets dans l'environnement, ce qui affecte aussi la qualité et la puissance du signal reçu. Beaucoup de modèles ont été proposés pour représenter cette atténuation. Ces modèles sont souvent paramétrés selon les conditions de l'environnement.

**Le modèle *Free space*** Le modèle le plus simpliste est le modèle *Free space*, aussi appelé le modèle de Friis [57]. Il suppose que le médium et les antennes sont parfaits en considérant que le signal parcourt uniquement le chemin en ligne de vue et qu'il n'y a aucun facteur de perturbation entre l'émetteur et le récepteur. Ce modèle considère donc que la portée est le rayon d'une sphère ou d'un cercle parfait, dans le plan de déplacement des entités, si le récepteur se situe à l'intérieur du cercle de l'émetteur il recevra le paquet sinon le paquet ne sera pas détecté.

**Le modèle *Two-ray ground*** Le modèle *Two-ray ground* considère que le signal est reçu selon deux chemins : le chemin direct et une réflexion sur le sol. Dans [58], il est prouvé que le modèle *Two-ray ground* est plus précis que celui de *Free space* pour des grandes distances, alors que c'est le contraire pour les petites distances, à cause des oscillations causées par la combinaison constructive et destructive des deux signaux.

**Le modèle *Shadowing*** Les modèles *Free space* et *Two-ray ground* supposent que la portée forme un cercle parfait en considérant la moyenne des puissances reçues à une distance donnée. En revanche, le modèle *Shadowing* [58] rajoute une composante aléatoire pour caractériser l'effet de l'environnement sur la propagation du signal. Ainsi, l'établissement d'une communication en limite de portée devient probabiliste et se rapproche de la réalité.

**Le modèle ITU-R P1238-4** L'ITU (*International Telecommunication Union*) est un organisme qui met à disposition des exploitants de liaisons radio des outils dédiés à la maîtrise de cette technologie. Le modèle ITU-R P1238-4 est conçu pour estimer l'atténuation du signal dans un milieu clos [59]. Ce modèle a été comparé dans [60] à des résultats expérimentaux issus de mesures faites dans un milieu industriel confiné. Ces résultats prouvent que le modèle ITU-R P1238-4 est plus proche de la réalité que les modèles *Free space*, *Two-ray ground* et *Shadowing*.

**Choix et vérification du modèle de propagation** Nous avons choisi d'utiliser le modèle ITU-R P1238-4 puisqu'il se rapproche le plus d'une modélisation réelle d'un environnement industriel confiné. Le modèle ITU-R P1238-4 n'existe pas par défaut parmi les outils ou options du simulateur NS2. Pour cela, nous avons dû l'implémenter et vérifier son fonctionnement en le comparant à des mesures.

Le modèle ITU-R P1238-4 estime la puissance reçue par la formule suivante :

$$\begin{aligned} Pr_{mW} &= \exp\left(\frac{Pt_{dBm} - L}{10} \log 10\right) / 1000, \\ L &= 20 \log_{10}(f) + 10 * N \log_{10}(d) + pfn - 28 + random, \end{aligned}$$

où  $Pr_{mW}$  est la puissance reçue  $mW$ ,  $Pt_{dBm}$  est la puissance de transmission en  $dBm$ ,  $L$  est l'atténuation ou le *Path Loss*,  $f$  est la fréquence en  $MHz$ ,  $N$  est le coefficient d'atténuation,  $d$  est la distance en mètre,  $pfn$  est le facteur d'atténuation qui prend en compte le nombre d'étages, dans notre cas il a été fixé à 0 en considérant que tous les nœuds se trouvent sur le même étage, *random* est une variable aléatoire qui suit une loi normale avec une moyenne de 0 et un écart type qu'on a fixé à 2.

Nous avons réalisé une série de mesures à l'intérieur d'un bâtiment, durant les heures de travail, ce qui nous permet de prendre en compte la mobilité des gens qui évoluent sur le site de mesures. Avec les paramètres du modèle ITU-R P1238-4 fixé comme spécifié précédemment, les résultats que nous avons obtenus sont donnés sur la figure 4.1. Nous avons fixé la puissance d'émission à  $-25$  dBm et nous nous sommes limités dans la simulation au seuil de sensibilité des mesures qui est à  $-95$  dBm. Nous pouvons remarquer qu'un grand nombre de points se superpose, ce qui prouve que le modèle de propagation implémenté dans notre simulateur émule d'une manière assez réaliste l'effet d'un environnement clos sur les signaux transmis. Nous avons préféré tolérer que notre modèle soit plus pessimiste que la réalité au lieu qu'il soit plus optimiste, ceci correspond en particulier à ce qui est observable sur les distances comprises entre 9 et 14 mètres. Notons aussi que durant les mesures, les antennes n'étaient pas toujours en ligne de vue.

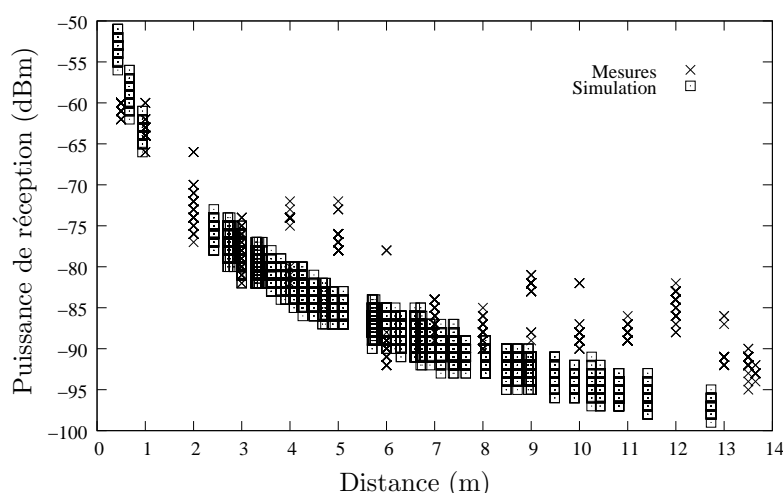


FIG. 4.1 – Comparaison entre les résultats des mesures faites en intérieur et les résultats du modèle ITU-R P1238-4.

Nous ne sommes pas allés plus loin dans notre étude du comportement du signal, car ceci sort du cadre de cette thèse.

### 1.1.2 Simulation de la couche physique

Un module radio ne peut détecter les informations reçues que d'une seule transmission à la fois. Le signal correspondant doit être reçu avec une puissance supérieure au bruit ambiant. Au bruit ambiant s'additionne la puissance reçue de tous les signaux détectés sur le même canal.

**Effet de capture et collision** Collision et capture sont deux conséquences complémentaires qui résultent de la réception, sur un récepteur donné, de plusieurs signaux émis dans des intervalles de temps ayant une partie commune. Cette réception simultanée de plusieurs signaux est à considérer au

niveau de chaque récepteur potentiel et selon des conditions de puissance il en résulte :

- soit une collision si aucun des signaux ne peut être convenablement interprété,
- soit une capture si le signal reçu dominant peut être compris.

Quand une station reçoit plusieurs signaux en même temps, différentes situations peuvent se produire. Si la différence entre la puissance reçue d'un signal et la somme des puissances reçues de tous les autres signaux est supérieure à un certain seuil, ce signal sera décodé et les autres signaux seront ignorés. Ce seuil est appelé seuil de capture. Si aucune des transmissions n'est vraiment dominante, tous les signaux sont perdus et il s'agit d'une collision <sup>2</sup>.

Durant la réception d'un signal, le module radio est synchronisé sur la modulation de ce signal, ceci lui donne un avantage quand un nouveau signal arrive avant la fin de la réception.

Pour prendre en compte ce phénomène de synchronisation, nous avons considéré deux seuils de capture différents : (i) un seuil dans le cas où le module radio n'est pas en cours de réception *seuil\_capture* et (ii) un seuil quand c'est le cas contraire pour favoriser le signal en cours de réception *seuil\_capture\_en\_cours*. L'algorithme 6 explique les étapes de test qu'une trame doit franchir avant de décider si elle est décodable ou non.

Nous avons choisi d'attribuer un tableau de trames à chaque récepteur. Ce tableau contient les trames en cours de transmission sur le canal. Chaque élément du tableau contient une copie de la trame, sa puissance reçue par la station, l'instant de fin de transmission et un booléen indiquant si la trame est décodable ou non.

Quand une nouvelle trame est reçue, elle est stockée dans ce tableau. La fin de transmission est calculée selon le débit du module radio et la longueur de la trame. Sa puissance de réception est calculée selon le modèle ITU et la distance qui sépare la station de l'émetteur. Si cette puissance est inférieure au seuil de réception, la trame est considérée non-décodable. Sinon, nous vérifions si une autre trame est en cours de réception. Si c'est le cas, la puissance reçue de la nouvelle trame est cumulée avec les puissances reçues de toutes les autres trames en cours de transmission. Si la différence entre cette somme et la puissance reçue de la trame en cours de réception est supérieure au *seuil\_capture\_en\_cours* la trame en cours de réception est considérée comme non-décodable, sinon, la réception en cours continue sa vie. Si la réception en cours est devenue non-décodable, encore faut-il comparer la puissance reçue de la nouvelle trame à la somme des puissances reçues des trames en cours de transmission, si la différence est supérieure au *seuil\_capture*, la nouvelle trame sera décodée, sinon, la nouvelle trame est considérée non-décodable.

---

<sup>2</sup>Nous distinguons deux types de capture : une capture productive c'est le cas où la trame rescapée concerne la station qui l'a capturée, et une capture improductive c'est le cas où la trame rescapée n'est pas destinée à la station qui l'a capturée.

À l'instant de fin de transmission, les trames qui ont été considérées non-décodables sont ignorées et toute trame décodable est remontée à la couche supérieure. À noter qu'une fois la trame est considérée non-décodable, elle est conservée dans le tableau pour contribuer au bruit ambiant jusqu'à la fin de sa transmission.

---

**Algorithm 6** Arrivée d'une nouvelle trame.

---

**Pré-requis :** un tableau de trames *tab*

**Pré-requis :** nouvelle trame *t*

Ajouter les informations concernant la nouvelle trame dans *tab*

**si** puissance reçue de *t* < *seuil\_de\_reception* **alors**

*t* est non-décodable

**sinon**

**si** réception en cours **alors**

*puissance\_recue\_totale* ← la somme des puissances reçues des trames de *tab* (hormis la trame en cours de réception)

**si** *puissance\_recue\_totale* > puissance reçue de la trame en cours de réception + *seuil\_capture\_en\_cours* **alors**

        la trame en cours n'est plus décodable

*puissance\_recue\_totale* = la somme des puissances reçues des trames de *tab* (hormis la nouvelle trame)

**si** puissance reçue de *t* < *puissance\_recue\_totale* + *seuil\_capture\_en\_cours* **alors**

*t* est non-décodable

**sinon**

*t* est décodable

**sinon**

        la trame en cours de réception continue sa réception

**fin si**

**fin si**

**sinon**

**si** transmission en cours **alors**

*t* est non-décodable

**sinon**

*puissance\_recue\_totale* = la somme des puissances reçues des trames de *tab* (hormis la nouvelle trame)

**si** puissance reçue de *t* < *puissance\_recue\_totale* + *seuil\_capture* **alors**

*t* est non-décodable

**sinon**

*t* est décodable

**fin si**

**fin si**

**fin si**

**fin si**

---

**L'implémentation du CCA** Selon la norme IEEE 802.15.4, il suffit que le CCA couvre une durée de 128  $\mu$ s (qui correspond à la durée d'envoi de 4 octets pour un débit de 250 *kbit/s*) pour scruter l'état du canal durant la première période de *backoff* des deux périodes de *backoff* consécutives prévues [38]. NS2 étant un simulateur événementiel, nous avons choisit d'émuler ce test qui dure sur un intervalle de temps de 128  $\mu$ s, par une séquence de tests instantanés.

Quand la couche MaCARI demande à la couche physique d'effectuer un CCA. Cette dernière teste

le canal en vérifiant s'il y a un signal avec une puissance reçue supérieure au seuil de réception (qui correspond à  $-95\text{ dBm}$ ). Ce test est répété toutes les  $16\text{ }\mu\text{s}$  tant que la puissance détectée ne dépasse pas ce seuil. Si au bout du huitième test la puissance détectée est toujours inférieure au seuil, la couche physique retourne un résultat indiquant que le canal est libre. Sinon, si un des tests détecte un signal au dessus du seuil, la couche physique remonte un résultat indiquant que le canal est occupé.

### 1.1.3 Simulation de la couche MAC

**L'implémentation du CSMA/CA slotté** La version du module CSMA/CA slotté qui existe sur NS2 est couplée aux modules des couches MAC et physique de la norme IEEE 802.15.4. De ce fait, il n'est pas utilisable avec notre couche MaCARI.

Le CSMA/CA est utilisé par MaCARI dans deux intervalles de temps différents : (i) l'intra-étoile pour les échanges entre les feuilles et les coordinateurs dans  $[T_1; T_2]$  et (ii) le routage entre les coordinateurs dans  $[T_2; T_3]$ . Les trames à envoyer dans ces deux intervalles ne sont pas les mêmes et la concurrence dans les deux intervalles est complètement indépendante. Ainsi, les paramètres du CSMA/CA utilisé dans  $[T_1; T_2]$  doivent être indépendants de ceux du CSMA/CA utilisé dans  $[T_2; T_3]$ . Pour ce faire, deux modules de CSMA/CA slotté différents sont créés pour chaque nœud et évoluent indépendamment dans deux intervalles de temps différents.

L'implémentation du module CSMA/CA slotté respecte l'algorithme décrit dans la partie 2.2.1 du chapitre 2. L'algorithme 7 décrit le déroulement du CSMA/CA slotté après le tirage de *backoff*. Le temps est décompté en période de *backoff* de  $320\text{ }\mu\text{s}$ . Le module est initié par la couche MaCARI au début de l'intervalle de temps en initialisant un compteur qui est décrémenté toutes les  $320\text{ }\mu\text{s}$ . Il est nécessaire de vérifier qu'il est possible d'émettre en prenant en compte la fin de l'intervalle pour empêcher l'entité d'accéder au médium au-delà de l'intervalle grâce aux mécanismes de report de CCA et de report de *backoff*.

Pour envoyer une trame en utilisant le CSMA/CA slotté, MaCARI passe la trame au module de CSMA/CA slotté concerné après en avoir gardé une copie. Cette copie est utilisée dans le cas des reports et des répétitions, et elle est supprimée quand la trame est envoyée. Le module CSMA/CA slotté commence par un tirage de *backoff*, puis il vérifie si le temps restant de l'intervalle de CSMA/CA slotté est suffisant pour consommer le *backoff*. Si c'est le cas, l'algorithme 7 s'exécute. Sinon, la trame est reportée à l'intervalle équivalent du cycle global suivant. Ainsi, l'algorithme 7 commence avec un *backoff* positif et est exécuté toutes les  $320\text{ }\mu\text{s}$ .



---

**Algorithm 7** L'algorithme de CSMA/CA slotté.

---

```

si fin de la période de csmaca = 0 alors
  return SLOTTED_CSMACA_ENDED
sinon
  décrémenter la fin de la période de csmaca
  si une trame est en attente d'envoi alors
    vérifier dans quel état on se trouve : (1) backoff > 0, (2) CW > 0, (3) transmission en cours
    si (1) alors
      décrémenter le backoff
      si backoff expiré alors
        passer en état (2)
      fin si
    sinon
      si (2) alors
        si il reste du temps pour faire 2 CCA, finir la transmission et recevoir un ACK alors
          effectuer un CCA
          si canal libre alors
             $CW \leftarrow CW - 1$ 
          sinon
            si NB > 5 alors
              tirer un nouveau backoff
            sinon
              return FAILURE_NB_EXCEEDED
            fin si
          fin si
        sinon
          return CCA_DEFERRED
        fin si
      sinon
        commencer la transmission
      fin si
    sinon
      si transmission en cours alors
        décrémenter la durée de transmission
        si transmission terminée alors
          return SUCCESS
        fin si
      fin si
    fin si
  fin si
fin si

```

---

#### 1.1.4 Simulation de la couche réseau

Dans le cadre du projet OCARI, la couche réseau est composée de deux protocoles : un protocole de routage EOLSR et un protocole de coloriage SERENA (comme indiqué en 1.2.1 et 1.2.2 du chapitre 3). Ces deux protocoles ont été proposés par le partenaire INRIA. Des problèmes de compatibilité et de disponibilité entre les codes de simulations développés par l'INRIA d'une part et le LIMOS d'autre part, ont rendu délicate la tâche de la fusion de ces deux codes. Le découpage temporel de MaCARI laisse la place pour un protocole de routage dans la période  $[T_2 ; T_3]$ . Néanmoins, pour évaluer MaCARI il nous a fallu développer un protocole de routage qui gère les échanges durant la période  $[T_2 ; T_3]$  et qui ne se limite pas aux liens de l'arbre.

Nous avons optimisé le protocole de routage hiérarchique, utilisé pour faire le relais dans  $[T_1 ; T_2]$ , en utilisant une table de voisinage qui permet de court-circuiter l'arbre dans certains cas.

**Protocole de routage  $[T_2 ; T_3]$**  Durant la période  $[T_2 ; T_3]$  tous les coordinateurs sont actifs et participent à l'acheminement des messages entre eux vers les destinations finales. Le protocole de routage utilisé durant cette période est basé sur le protocole de routage hiérarchique de ZigBee et la table de voisinage. Cette table de voisinage est construite durant la période  $[T_0 ; T_1]$ , elle comprend l'adresse courte du coordinateur père, des coordinateurs fils et de chaque coordinateur duquel un *beacon* est reçu.

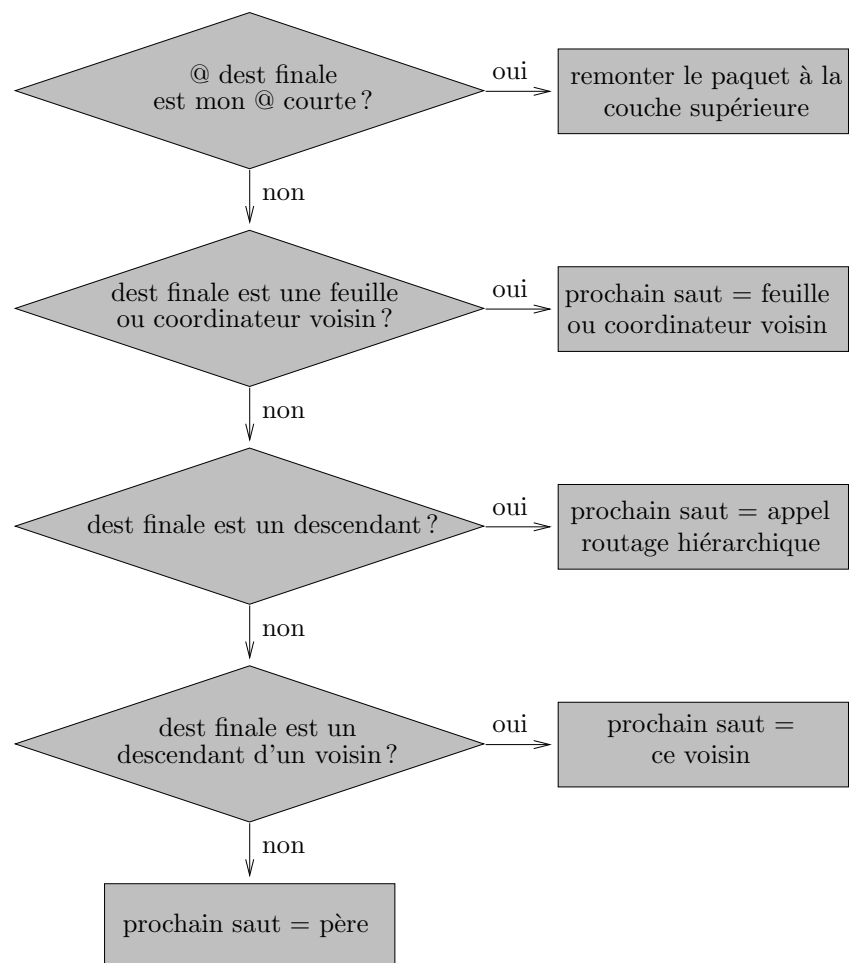
Le diagramme présenté sur la figure 4.2 décrit l'algorithme de routage utilisé durant  $[T_2 ; T_3]$ . La décision du prochain saut est prise en fonction de l'adresse courte de la destination finale du paquet.

Si l'adresse destination finale est l'adresse courte du coordinateur, le paquet est remonté à la couche supérieure. Si cette adresse correspond à une adresse courte d'une feuille du coordinateur ou d'un coordinateur voisin, l'adresse du prochain saut sera l'adresse de la destination finale. Sinon, si la destination finale est parmi les descendants, l'adresse du prochain saut est un des fils coordinateur calculée selon la formule donnée dans 2.3.3. Sinon, si un des voisins, qui n'est ni un fils ni le père, a la destination finale dans sa descendance, le prochain saut sera l'adresse de ce voisin. Sinon, le paquet est remonté au père.

Ce protocole ne demande aucun échange supplémentaire ni une gestion de table de routage, il ne surcharge donc pas le réseau par un trafic de contrôle. Toutes les informations nécessaires sont déjà récupérables grâce à la couche MaCARI.

#### 1.1.5 Aspect graphique

Nous avons travaillé dans l'équipe sur le développement d'un animateur graphique, MaCAview, qui illustre les fonctionnalités de MaCARI et nous permet de suivre l'activité du réseau durant les différents intervalles de temps. MaCAview prend en entrée le fichier de trace qui enregistre tous les envois et les réceptions générés par le scénario de simulation, et des informations spécifiques à MaCARI comme les

FIG. 4.2 – Diagramme de l'algorithme de routage  $[T_2; T_3]$ .

durées des différentes périodes du cycle global et les liens de parenté, et nous sorte un scénario animé illustrant l'activité du réseau.

La figure 4.3 nous montre une capture d'écran de MaCAview. Nous pouvons identifier les liens de parenté, l'intervalle de temps dans lequel on se trouve (cette capture montre qu'on se trouve durant la période de synchronisation grâce à l'indication T0), des informations concernant la trame en cours de transmission comme la taille et la source, ainsi que des informations concernant les nœuds comme l'adresse courte et l'adresse longue. Au moment de la capture, le CPAN était en cours de diffusion du beacon. Nous avons choisi de représenter une diffusion par la transmission de plusieurs trames (une trame par récepteur concerné).

Cet animateur nous a été utile, notamment pour déboguer notre code de simulation sans avoir besoin de lire et de décortiquer les fichiers de traces qui peuvent dépasser les 10 *MO* de taille.

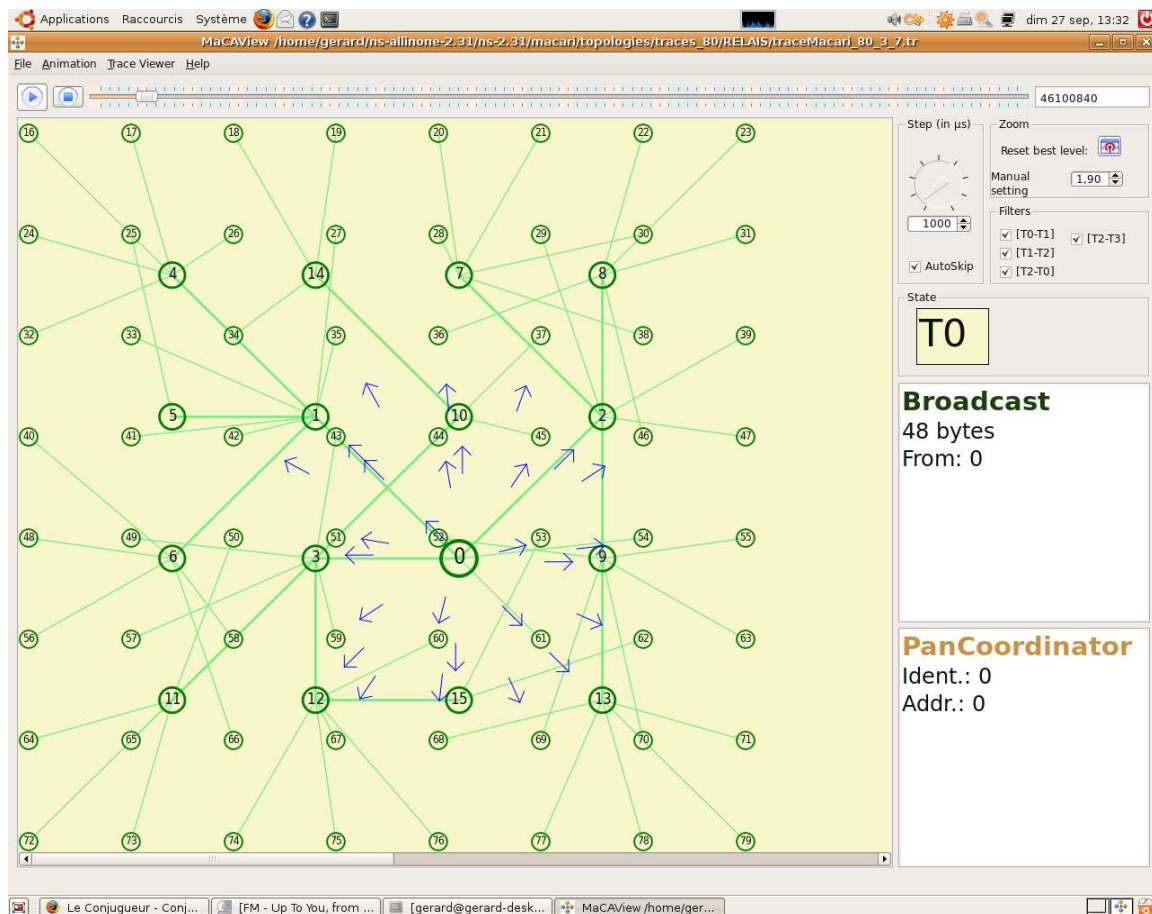


FIG. 4.3 – Capture MaCAview.

## 1.2 Maquettage

Le rôle de cette étape est d'une part d'être confronté aux limites du matériel, i.e. le module radio, le microcontrôleur et ses capacités, et d'une autre part de satisfaire à l'obligation de fournir un prototype dans le cadre du projet OCARI.

Cette maquette a pour rôle de montrer les fonctionnalités essentielles de MaCARI, i.e. la synchronisation, le découpage temporel, le relais garanti, et de prouver leur faisabilité.

MaCARI étant un protocole MAC, il interagit avec la couche physique et la couche réseau. La couche physique adoptée est celle de la norme IEEE 802.15.4. Notre partenaire Telit nous a fourni cette partie du code. La planification des différentes étapes du projet OCARI est telle que la disponibilité de la couche réseau est prévue après la réalisation de la maquette. Pour montrer d'une part le bon fonctionnement des échanges entre MaCARI et la couche réseau, et d'autre part l'intérêt du découpage en 4 périodes de temps, il nous a fallu disposer d'une couche 3 provisoire qui communique avec MaCARI et qui applique un protocole de routage pour acheminer les paquets durant la période  $[T_2; T_3]$ . Ainsi, nous avons développé, conjointement avec notre partenaire LATTIS, une interface homme-machine (IHM) exécutée sur un ordinateur, qui communique avec la carte sur laquelle nous avons développé MaCARI à travers un port série. Le rôle de cette IHM est d'effectuer les tâches suivantes :

- Appliquer un protocole de routage pour le trafic échangé durant  $[T_2; T_3]$ ,
- Envoyer des requêtes d'envoi à la couche MaCARI,
- Afficher la réception de trames chez la destination finale,
- Valider les échanges de primitives entre la couche réseau et la couche MaCARI.

Le consortium OCARI a défini l'envergure d'une maquette limitée à une quinzaine de cartes. La topologie de cette maquette est représentée sur la figure 4.4. La photo 4.5 montre les cartes sur lesquelles nous avons développé, ainsi que l'IHM sur l'ordinateur portable qui apparaît dans la photo.

### 1.2.1 Implémentation sur des cartes B2400ZB

Notre plateforme de développement est constituée de cartes B2400ZB-tiny [13] composées d'un module radio Chipcon CC2420 [61] et d'un microprocesseur Motorola MC9S08GT60A [62]. Cette solution était le choix sur lequel le consortium OCARI s'était mis d'accord. Le développement a été fait en C à l'aide de l'outil de développement croisé CodeWarrior [63].

### 1.2.2 Synchronisation et compteurs

Grâce à la synchronisation, les nœuds du réseau partagent une référence temporelle qui est  $T_1$ . À  $T_1$ , un compteur est initialisé et incrémenté toutes les  $320 \mu s$ . Ce compteur nous permet de gérer le

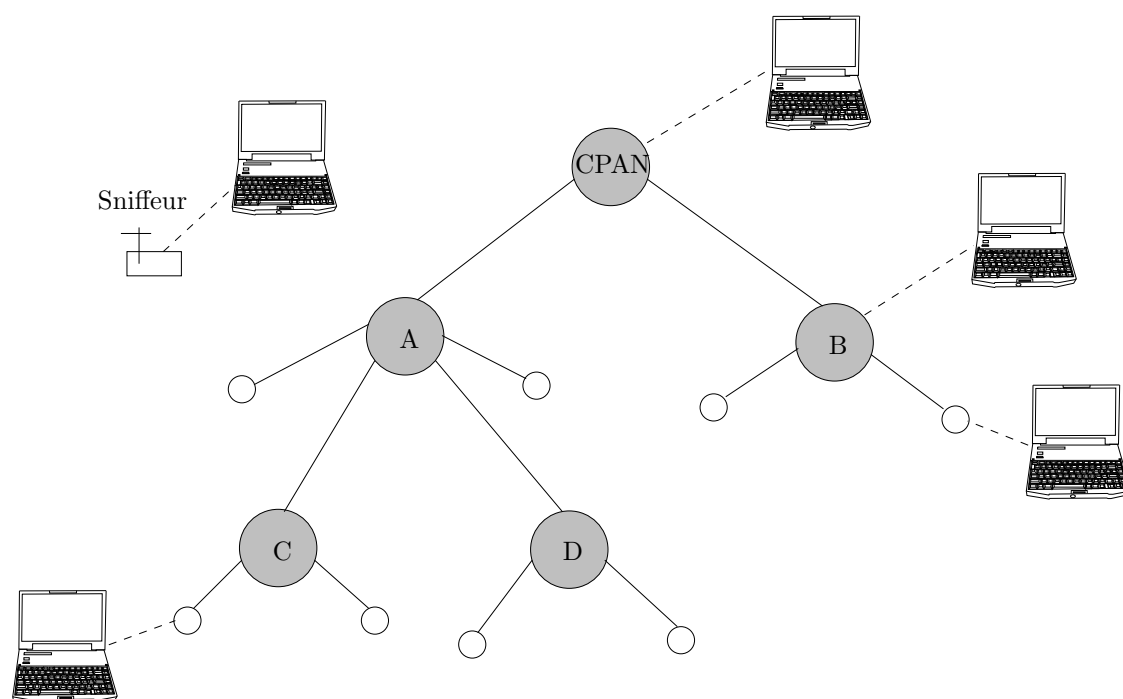


FIG. 4.4 – La topologie de la maquette dans le cadre du projet OCARI.

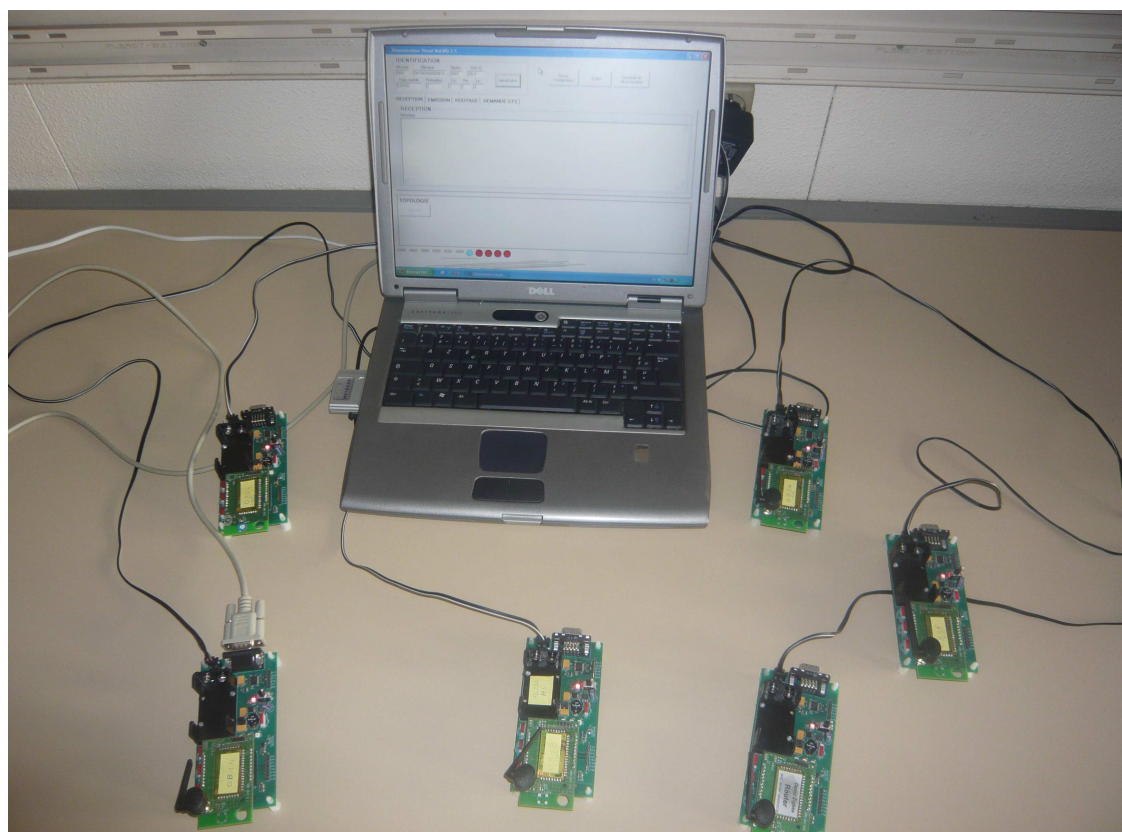


FIG. 4.5 – Une photo des cartes utilisées pour la maquette.

découpage temporel de MaCARI. Comme les durées des intervalles de temps sont déduites du contenu du *beacon* et codées à partir d'une unité égale à  $320 \mu s$ , chaque entité du réseau sait dans quel intervalle il se trouve en surveillant ce compteur et en fonction des durées des intervalles qui le concernent.

### 1.2.3 Machine d'état MaCARI et gestion des files d'attente

Le fonctionnement du nœud dépend de l'intervalle de temps dans lequel il se trouve. Ce fonctionnement est représenté dans la figure 3.13 sous forme d'une gestion d'activité selon l'intervalle de temps dans lequel se trouve le nœud. Pour des raisons de limitation en taille mémoire des cartes sur lesquelles nous travaillons, nous avons choisi de gérer uniquement deux files d'attente différentes : (i) une file d'attente pour la période  $[T_1; T_2]$  qui contient les trames à envoyer dans l'étoile et les trames à relayer durant les intervalles de relais et (ii) une file d'attente pour la période  $[T_2; T_3]$  qui contient les trames, de type trafic non-prioritaire, à échanger entre coordinateurs.

À la réception d'une trame, la couche MaCARI vérifie son type et la met dans la file d'attente correspondante. Le passage d'un intervalle à un autre permet au nœud de connaître quelle file d'attente il doit solliciter.

### 1.2.4 Gestion des reports

Quand le nœud détecte qu'il ne lui reste pas assez de temps pour envoyer une trame durant l'intervalle de temps courant, il garde cette dernière dans la file d'attente pour l'envoyer dans le prochain cycle global. En effet, cela peut arriver dans deux cas : (i) les trames dans la file d'attente n'ont pas pu être toutes envoyées ou (ii) une requête d'émission d'une trame arrive vers la fin de l'intervalle courant.

## 2 Synchronisation

Dans cette partie, nous allons évaluer en fonction de la taille du réseau la durée de la période de synchronisation par simulation et montrer le gain que nous offre l'amélioration par envoi anticipé (cf. 2.7.1 du chapitre 3). Ensuite, nous allons estimer la précision sur l'instant  $T_1$  calculé à partir de mesures faites sur les cartes B2400ZB-tiny.

### 2.1 Simulation

L'étape de simulation de la synchronisation nous permet de quantifier la durée de cette période et l'intérêt de la méthode d'envoi anticipé. Pour ce faire nous avons généré des topologies aléatoires qui respectent un paramétrage de topologie donné (en fonction de  $Rm$ ,  $Cm$  et  $Lm$ ).

#### 2.1.1 Scenarii de tests

Nous avons généré 100 topologies différentes avec une profondeur maximale de 3. Nous avons fait varier  $Rm$  de 3 à 5 pour chaque nombre de coordinateurs considéré. Les courbes pointillées représentent les moyennes des 100 valeurs données par nos répliques. Les segments verticaux représentent les écart-types.

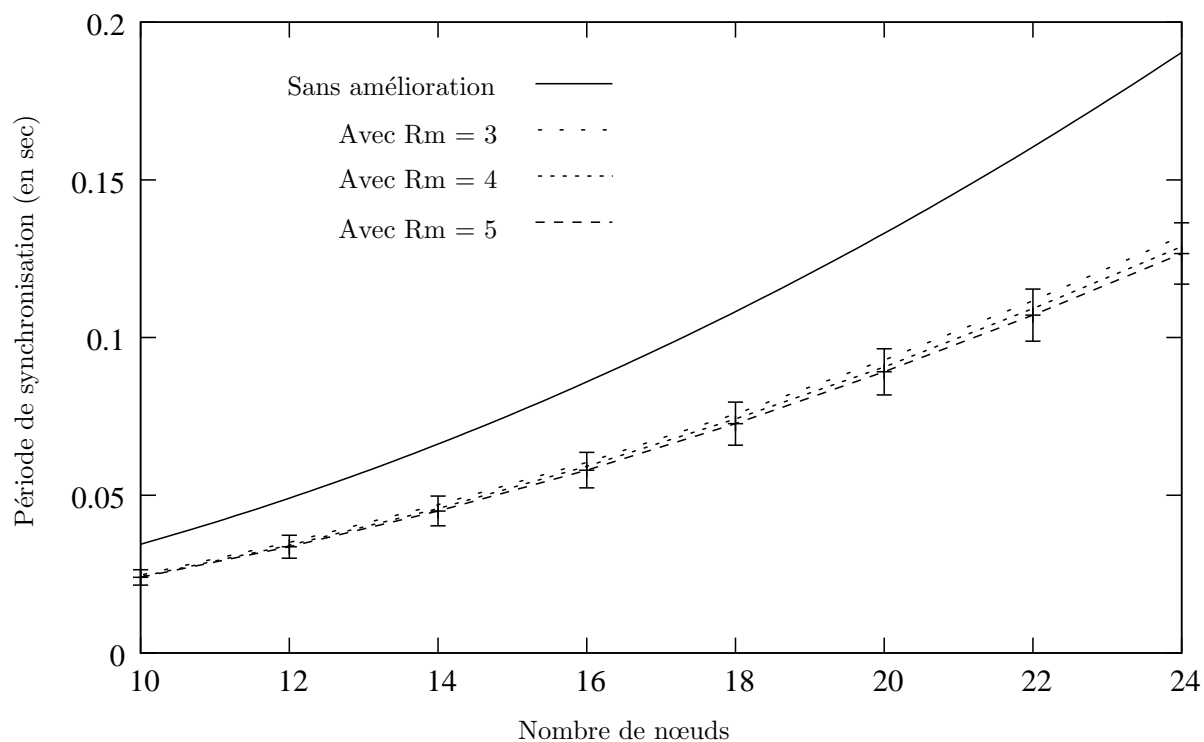


FIG. 4.6 – Effet de l'envoi par anticipation sur la durée de la période de synchronisation pour des configurations de 10 à 24 coordinateurs.

La figure 4.6 montre que la durée de la période de synchronisation augmente exponentiellement en



fonction du nombre de coordinateurs dans le réseau. Ceci est attendu car la taille du *beacon* augmente avec le nombre de coordinateurs. La courbe sans amélioration correspond au cas où la méthode d'envoi anticipé n'est pas appliquée. Nous pouvons aussi constater le gain obtenu grâce à l'amélioration par envoi anticipé. En faisant varier le nombre de fils maximum par coordinateur, les écarts-type nous montrent que le gain est peu sensible à ce paramètre.

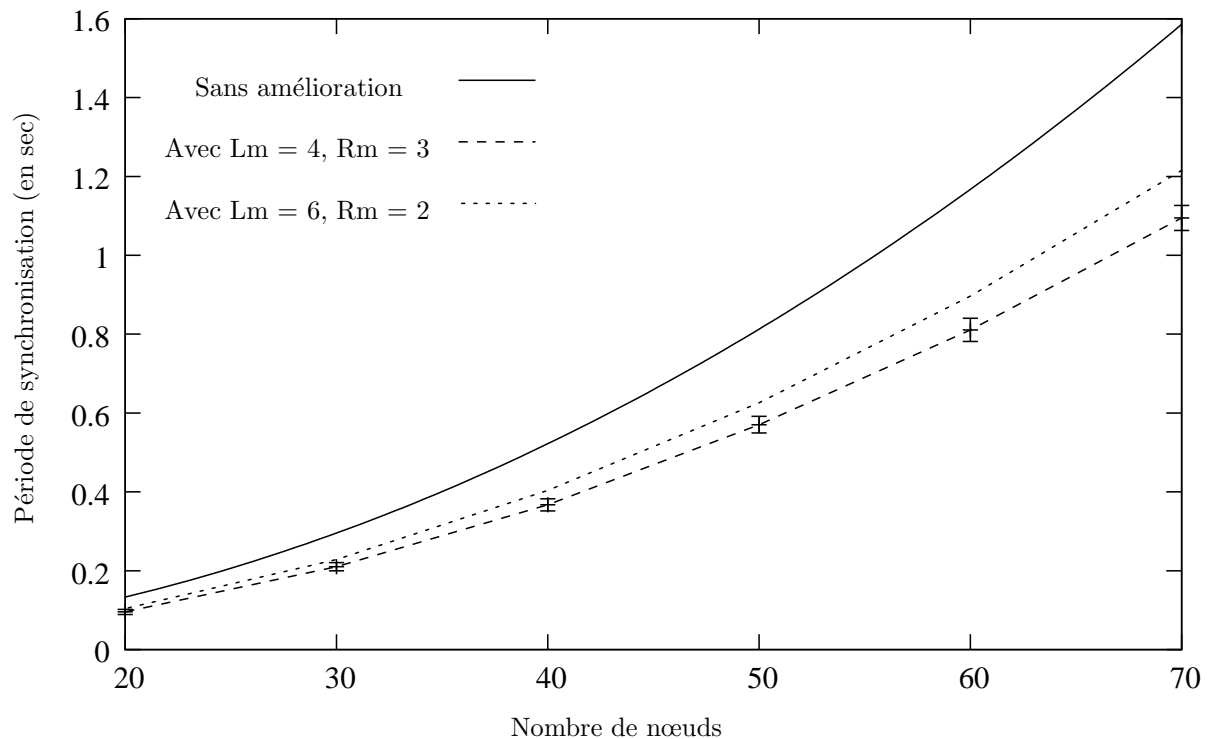


FIG. 4.7 – Effet de l'envoi par anticipation sur la durée de la période de synchronisation pour des configurations de 20 à 70 coordinateurs.

L'envoi par anticipation diminue significativement la durée de la période de synchronisation. La figure 4.7 montre que pour une topologie de 70 coordinateurs et avec  $L_m = 4$  et  $R_m = 3$ , la durée de  $[T_0; T_1]$  est réduite de 31,25%.<sup>3</sup>

## 2.2 Mesures de précision

Le découpage temporel de MaCARI est basé sur le calcul effectué durant la période de synchronisation. Il est primordial de prouver que la mécanique de synchronisation fonctionne sur les cartes B2400ZB-tiny et de mesurer la précision obtenue.

<sup>3</sup>Ces résultats ont été publiés dans [64].

### 2.2.1 Scénarii de tests

Nous avons construit une topologie uniquement avec des coordinateurs qui correspond à la configuration de la maquette. Cette topologie est représentée sur la figure 4.8.

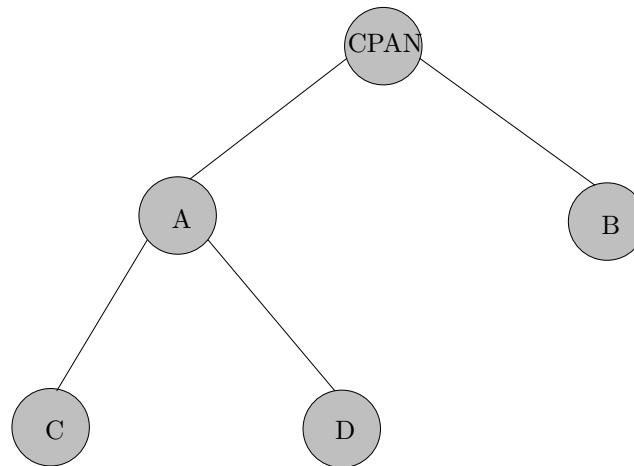


FIG. 4.8 – La topologie de test de précision de la synchronisation sur  $T_1$ .

En utilisant un analyseur d'états logiques (cf. figure 4.9), nous avons calculé le décalage de la synchronisation sur  $T_1$  entre les différentes entités. Sur 20 mesures, nous avons obtenu un décalage qui ne dépasse pas les  $640 \mu s$  sur la précision du calcul de  $T_1$ . Les moyennes, les écarts-type et les valeurs maximales de ce décalage sont donnés sur le tableau 4.1.

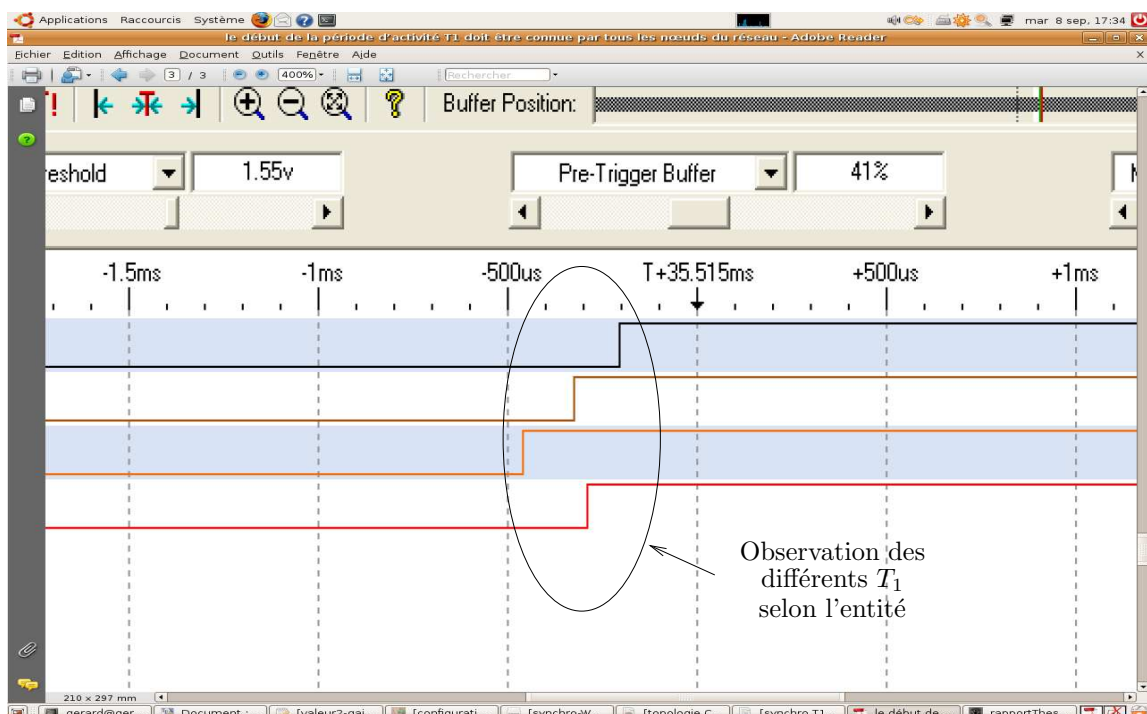


FIG. 4.9 – Un zoom sur les résultats de l'analyseur d'états logiques.

	<b>CPAN et A</b>	<b>CPAN et C</b>	<b>B et C</b>
Moyenne	119,55	338,95	303,65
Écart-type	88,83	101,36	178,35
Maximum	320	567	589

TAB. 4.1 – Précision sur le calcul de  $T_1$  en  $\mu s$ .

Ces résultats sont acceptables pour une maquette de faisabilité dans la mesure où une garde de 640  $\mu s$  (2 périodes de *backoff*) peut être mise en place pour éviter le chevauchement des périodes d'activité.

### 3 Segmentation

Dans ce chapitre, nous allons évaluer les performances de la méthode d'accès et l'amélioration apportée par la segmentation en différents intervalles de temps selon les trois types d'activité suivants : (i) la récolte des informations générées par les feuilles par le coordinateur d'étoile, (ii) le relais garanti du trafic de type prioritaire entre les couples coordinateurs père-fils et (iii) le routage non garanti entre les coordinateurs durant  $[T_2; T_3]$ .

Nous commençons par une approche simulation pour évaluer les performances du CSMA/CA slotté dans un intervalle de temps limité et pour montrer l'intérêt de la segmentation temporelle, en comparant 3 configurations différentes de MaCARI. Ensuite, nous évaluons le délai de bout-en-bout pour le trafic de type prioritaire.

Dans la seconde partie, nous reprenons la même démarche pour évaluer les performances du CSMA/CA slotté et le délai de bout-en-bout mesurés sur les cartes B2400ZB-tiny pour faire le rapprochement entre l'aspect simulation et maquettage.

#### 3.1 Simulation

Dans cette partie nous commençons par une évaluation du comportement de CSMA/CA slotté dans des périodes d'activité délimitées, en insistant sur l'effet du report de trames. En suite, nous comparons le comportement de CSMA/CA slotté avec celui d'un mécanisme de *polling* dans l'étoile. Nous étudions en suite l'amélioration que nous offre le découpage temporel de MaCARI en plusieurs intervalles d'activité. Nous terminons avec une quantification du délai de bout-en-bout.

##### 3.1.1 Évaluation de la durée des périodes d'activité

Pour dimensionner la durée des périodes d'activité de  $[T_1; T_2]$  du cycle global de MaCARI, il nous a fallu mesurer le temps nécessaire pour faire passer une certaine quantité de trafic en utilisant le CSMA/CA slotté comme méthode d'accès.

Pour ce faire, nous avons considéré plusieurs configurations d'étoiles en faisant varier le nombre de feuilles actives (i.e. qui ont quelques choses à dire) et la quantité de trafic que chaque feuille génère. Suite à la réception du *beacon*, les feuilles commencent à envoyer 1, 3 ou 5 trames chacune. Les trames sont à destination du coordinateur et elles font 50 octets chacune (cette longueur ayant été jugée la plus significative dans notre domaine par le consortium OCARI).

Dans un premier temps, nous avons quantifié le temps nécessaire pour que toutes les trames soient reçues par le coordinateur. La figure 4.10 montre que le temps augmente d'une manière logarithmique en fonction du nombre de feuilles et de la charge qu'elles génèrent. Chaque point correspond à une moyenne

de 100 répliques. Dans une étoile de 5 feuilles, où chaque feuille doit envoyer 3 trames, le temps nécessaire pour écouler ce trafic est de 75 ms.

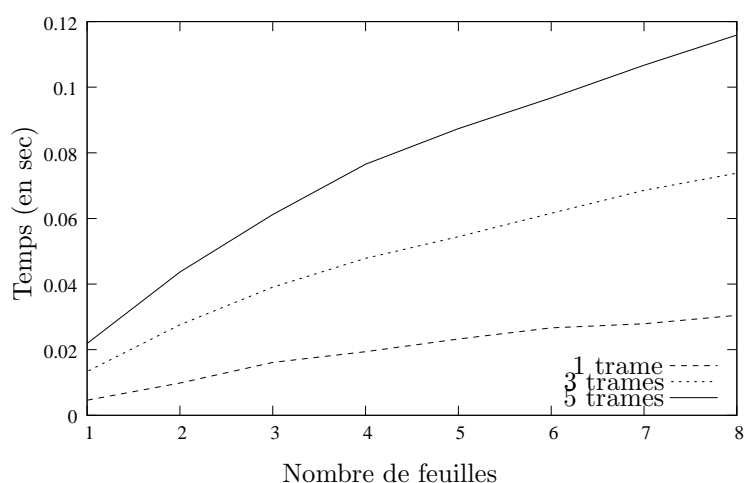


FIG. 4.10 – Simulation : Temps nécessaire pour écouler une certaine quantité de trafic dans une étoile en fonction du nombre de feuilles et du nombre de trames générées par feuille.

Ensuite, nous nous sommes intéressés à la quantification du nombre de trames reçues par le coordinateur. Sur la figure 4.11, nous pouvons voir le nombre de trames reçues par le coordinateur pour différentes charges de l'étoile. Chaque point correspond à une moyenne de 100 répliques.

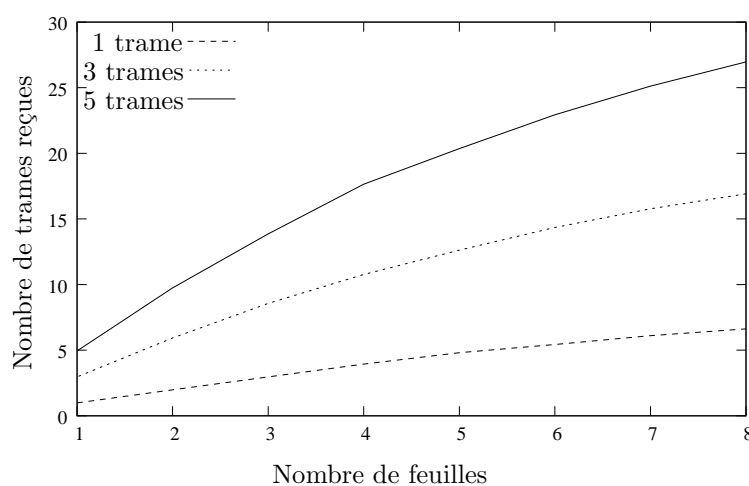


FIG. 4.11 – Simulation : Nombre de trames reçues par le coordinateur d'une étoile en fonction du nombre de feuilles et du trafic généré par chaque feuille.

Considérons une étoile ayant 8 feuilles actives. Prenons le cas où chaque feuille doit générer 3 trames. Le nombre total de trames devant être générées est donc 24, alors que 17 trames uniquement ont été reçues par le coordinateur de cette étoile, comme l'indique la courbe en pointillé fin de la figure 4.11. Cet écart est essentiellement dû aux collisions de trames. Suite à un certain nombre de transmissions sans réception d'acquittement, la couche MAC rejette la trame. Notons que sous faible charge le nombre de trames reçues par le coordinateur correspond à celui de trames générées, alors que sous forte charge les

pertes de trames montrent les effets d'une forme de saturation de l'étoile.

### 3.1.2 Évaluation du report du CSMA/CA slotté

Le CSMA/CA slotté est utilisé durant des périodes de taille délimitée, cela exige l'existence d'un mécanisme de report de trames pour éviter que des transmissions soient commencées vers la fin d'une période et chevauchent sur la période suivante. Selon l'algorithme de CSMA/CA slotté de la norme IEEE 802.15.4, avant d'envoyer une trame une station tire un *backoff* aléatoire. Si ce *backoff* est plus grand que la durée restante de la période d'activité (dans le cas de la norme IEEE 802.15.4 c'est la CAP, et dans MaCARI c'est la période d'activité d'une étoile dans  $[T_1; T_2]$  ou durant la période  $[T_2; T_3]$ ), il est décrementé jusqu'à la fin de la période en cours et le reste sera reporté à la période d'activité suivante. Un autre report peut avoir lieu aussi, il concerne le test du canal (CCA). Après avoir consommé le *backoff* la station vérifie si ce qui lui reste de temps dans la période d'activité est suffisant pour finir la transaction d'envoi : les 2 CCA, l'envoi de la trame physique et l'attente d'un éventuel acquittement. Si ce qui reste du temps n'est pas suffisant, la trame sera reportée pour être envoyée dans la période d'activité suivante.

Ce report est souvent ignoré par les études faites sur les performances du CSMA/CA slotté de la norme IEEE 802.15.4. Dans [65], [66], [67], [68] et [69] les auteurs n'ont pas considéré le report ni dans leur modèle analytique de CSMA/CA slotté, ni dans leurs simulations. Dans [70], les auteurs trouvent un taux élevé de collisions en début de la supertrame, ceci est aussi constaté par les auteurs dans [42]. Ces derniers concluent que cela est dû à un défaut dans la version 2003 de l'algorithme de CSMA/CA slotté [71]. Dans cette version, suite à un report, les nœuds ne tirent pas un nouveau *backoff*, ce qui induit des collisions systématiques en début de la prochaine supertrame. Néanmoins, ce défaut a été corrigé dans la version 2006 de la norme et les nœuds tirent un nouveau *backoff* en début de la supertrame après avoir subi un report.

**Le modèle de Markov de CSMA/CA slotté** Pour présenter l'ensemble des cas que provoque un report nous avons choisi d'utiliser les chaînes de Markov comme un outil de modélisation graphique. Nous allons commenter la représentation graphique que nous trouvons dans la littérature. Chaque état du modèle représente la consommation d'une période de *backoff*. La chaîne est composée de  $m$  étages de *backoff*. Chaque étage est constitué de  $W_m - 1$  états de décrémentation de *backoff*. Quand un *backoff*  $b$  est tiré à l'étage  $k$  on passe à l'état  $(k, b)$ . Les états visités par la suite sont  $(k, b-1)$ ,  $(k, b-2)$ ,  $\dots$ , jusqu'à  $(k, 1)$ . Quand l'état  $(k, 1)$  est visité ou bien quand un *backoff* de 0 est tiré, le premier CCA est effectué (ce qui correspond à  $(k, CCA_1)$ ). Suite à la scrutation du canal, un état occupé est constaté avec une probabilité  $\alpha$  et un état libre est constaté avec une probabilité  $1 - \alpha$ . Si le canal est occupé, un nouveau *backoff* est tiré dans une fenêtre plus grande (ce qui correspond à une incrémentation du  $BE$ ). Ainsi,

l'état suivant est l'étage  $k + 1$  de *backoff*. Si le canal est libre, un second CCA est effectué (représenté par  $(k, CCA_2)$ ). Le canal est occupé avec une probabilité  $\beta$  et libre avec une probabilité  $1 - \beta$ . Si le canal est occupé suite au second CCA, un nouveau *backoff* est tiré dans l'étage  $k + 1$ . Si le canal est libre, la transmission peut commencer.

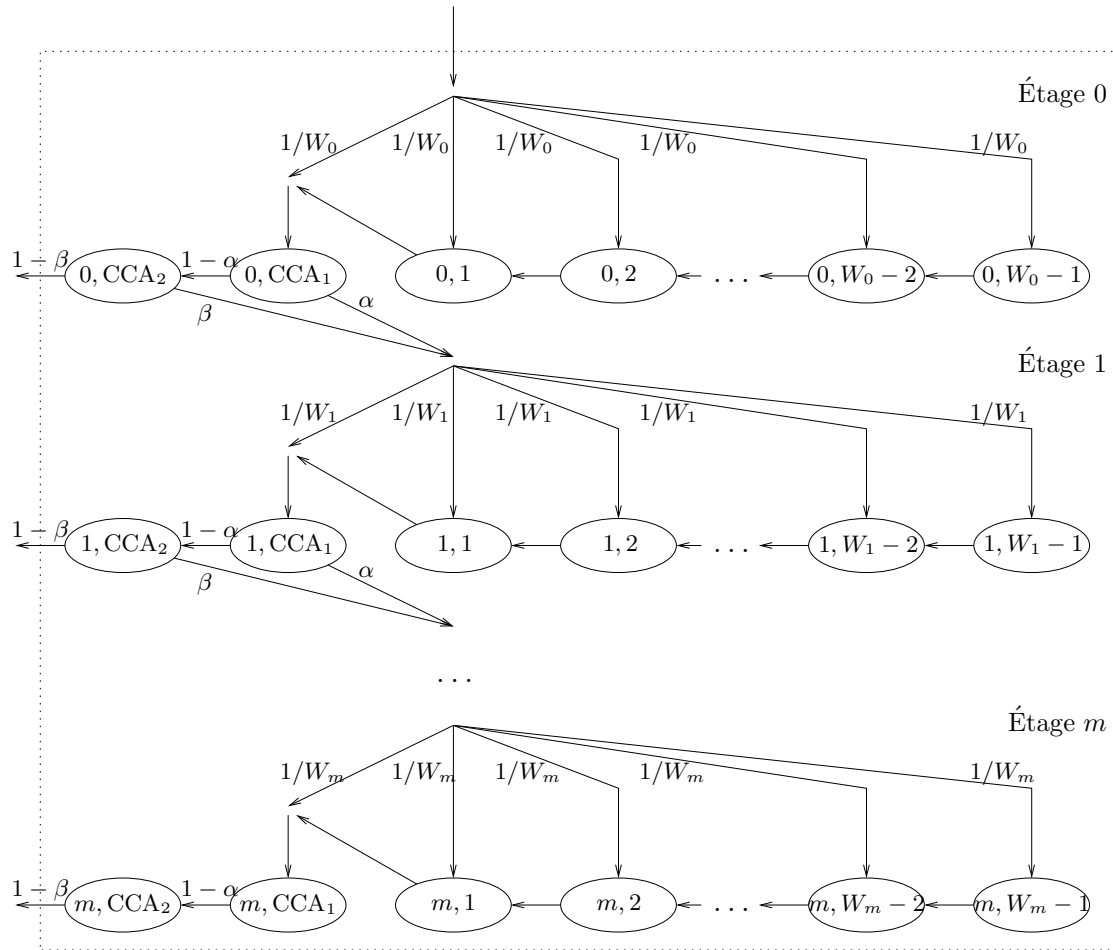


FIG. 4.12 – Le modèle de Markov du CSMA/CA slotté de la norme IEEE 802.15.4 sans les mécanismes de report.

Nous pouvons constater que les mécanismes de report (le report de *backoff* et le report de CCA) n'ont pas été inclus dans le modèle représenté sur la figure 4.12. Sur la figure 4.13, nous proposons une chaîne de Markov incluant les deux reports. Dans ce modèle, deux types de boîtes ont été ajoutées par rapport au modèle présenté sur la figure 4.12 : le report de CCA qui correspond à la boîte  $(k, \text{report}^C)$  et le report de *backoff* qui correspond à la boîte  $(k, \text{report}_b^B)$ .

Quand un *backoff* est tiré à un étage  $k$ , l'état suivant est  $(k, \text{report}_b^B)$  au lieu de passer directement à l'état  $(k, b)$ . Arrivant à la boîte  $(k, \text{report}_b^B)$ , il y a une probabilité  $\delta$  que la trame soit reportée à cause d'un report de *backoff* et une probabilité  $1 - \delta$  qu'elle ne le soit pas (c.f. figure 4.15). Le report de CCA risque d'avoir lieu juste avant d'effectuer les 2 CCA. À ce moment, il y a une probabilité  $\gamma$  que le nombre

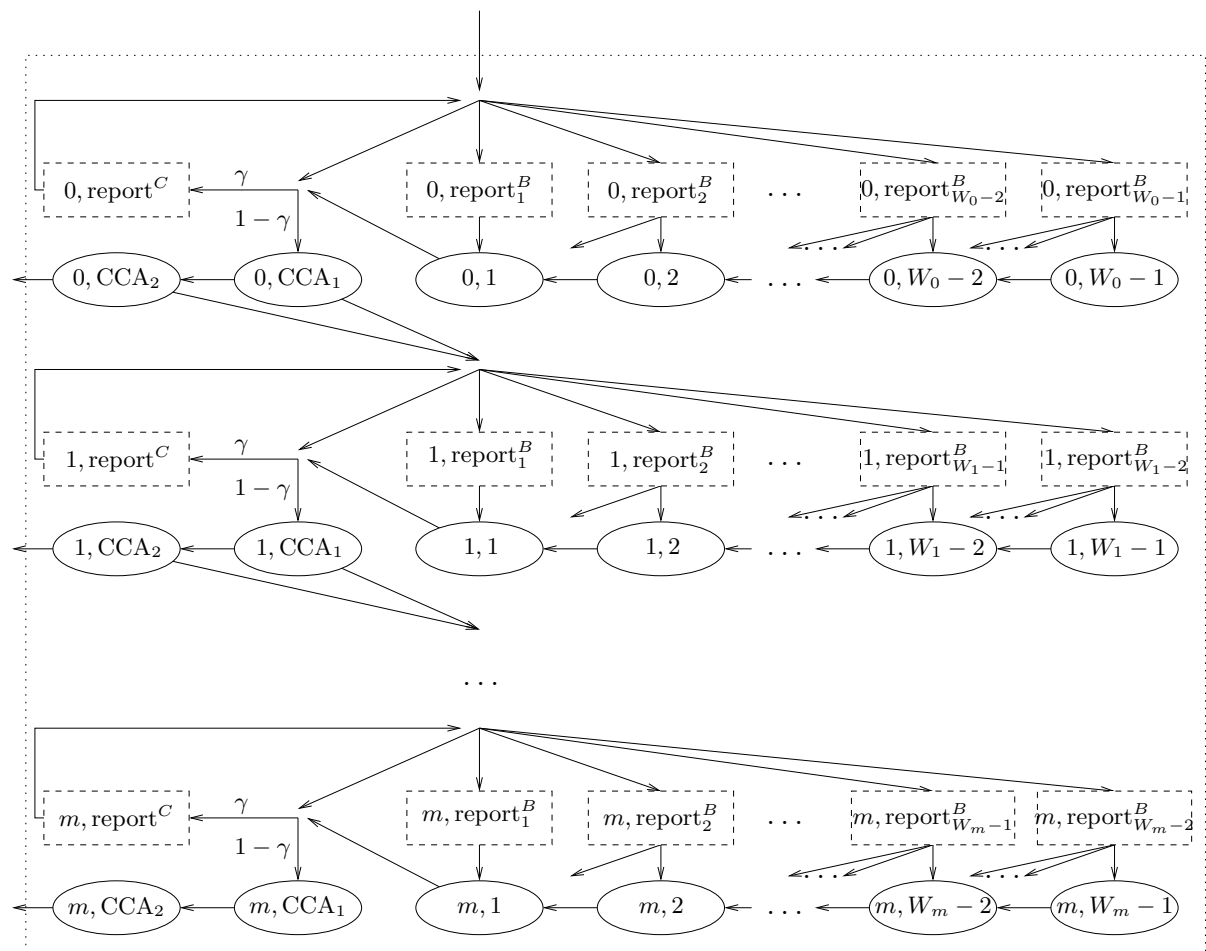


FIG. 4.13 – Chaîne de Markov représentant l'algorithme de CSMA/CA slotté de IEEE 802.15.4.



de périodes de *backoff* restant ne soit pas suffisant pour finir la transaction d'envoi, alors, l'état suivant est  $(k, \text{report}^C)$ . La probabilité d'avoir le temps de finir la transaction d'envoi est  $1 - \gamma$ , dans ce cas, l'état suivant est  $(k, CCA_1)$ .

La figure 4.14 détaille le contenu de la boîte  $(k, \text{report}^C)$  qui représente le report de CCA. Quand un report de CCA est effectué, il est nécessaire de savoir combien de période de *backoff* il faut attendre avant la fin de la période d'activité.

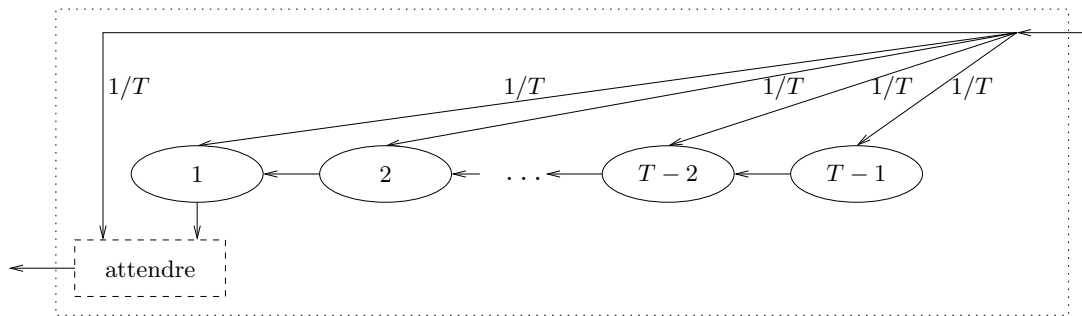


FIG. 4.14 – Report de CCA à la période d'activité suivante (boîte  $(k, \text{report}^C)$  de la figure 4.13).

Soit  $T$  le temps nécessaire pour effectuer la transaction d'envoi exprimé en périodes de *backoff* : les 2 CCA, l'émission de la trame et l'attente de la réception d'un éventuel acquittement. Ainsi, dans le cas d'un report de CCA, le nombre de périodes de *backoff* restant dans la période d'activité ne dépasse pas  $T - 1$ . Nous supposons à ce niveau que la probabilité de passer dans chacun des états est la même. Cela représente un réseau non saturé dans lequel la génération de trafic est uniforme vers la fin de la période d'activité. Nous considérons que les trames reportées ont été envoyées en début de la période d'activité et ainsi, elles n'affectent pas l'aspect uniforme de la génération de trafic. Une fois la période d'activité terminée, la station attend le début de la prochaine période d'activité. Cette attente correspond à la boîte "attendre".

La figure 4.15 représente la boîte  $(k, \text{report}_b^B)$  qui correspond au report d'un *backoff*. Cet état est atteint quand un *backoff*  $b$  est tiré à l'étage  $k$  de la figure 4.13.

La probabilité que  $b$  soit plus grand que le nombre de périodes de *backoff* est  $\gamma$  (la probabilité que ce soit plus petit est  $1 - \gamma$ ). Ainsi, la probabilité de ne pas avoir à effectuer un report de *backoff* est  $1 - \gamma$  et l'état suivant dans la chaîne de Markov est  $(k, b)$ . Quand il y a un report de *backoff*, il est nécessaire de savoir combien de période de *backoff* il faut attendre avant la fin de la période d'activité. Nous retenons la même hypothèse que celle du report de CCA pour calculer la probabilité de chacun des états  $(k, x, x)$ . Ceci nous donne une probabilité de  $1/b$  pour atteindre un état  $(k, x, x)$ . Un état  $(k, x, y)$  signifie :  $x$  est le nombre de périodes de *backoff* restant dans la période d'activité en cours quand l'état  $(k, x, x)$  est atteint,  $y$  est le nombre de périodes de *backoff* à consommer avant la fin de la période d'activité en cours. En sortant de l'état  $(k, x, 1)$ , la station doit attendre le début de la période d'activité suivante, ce qui

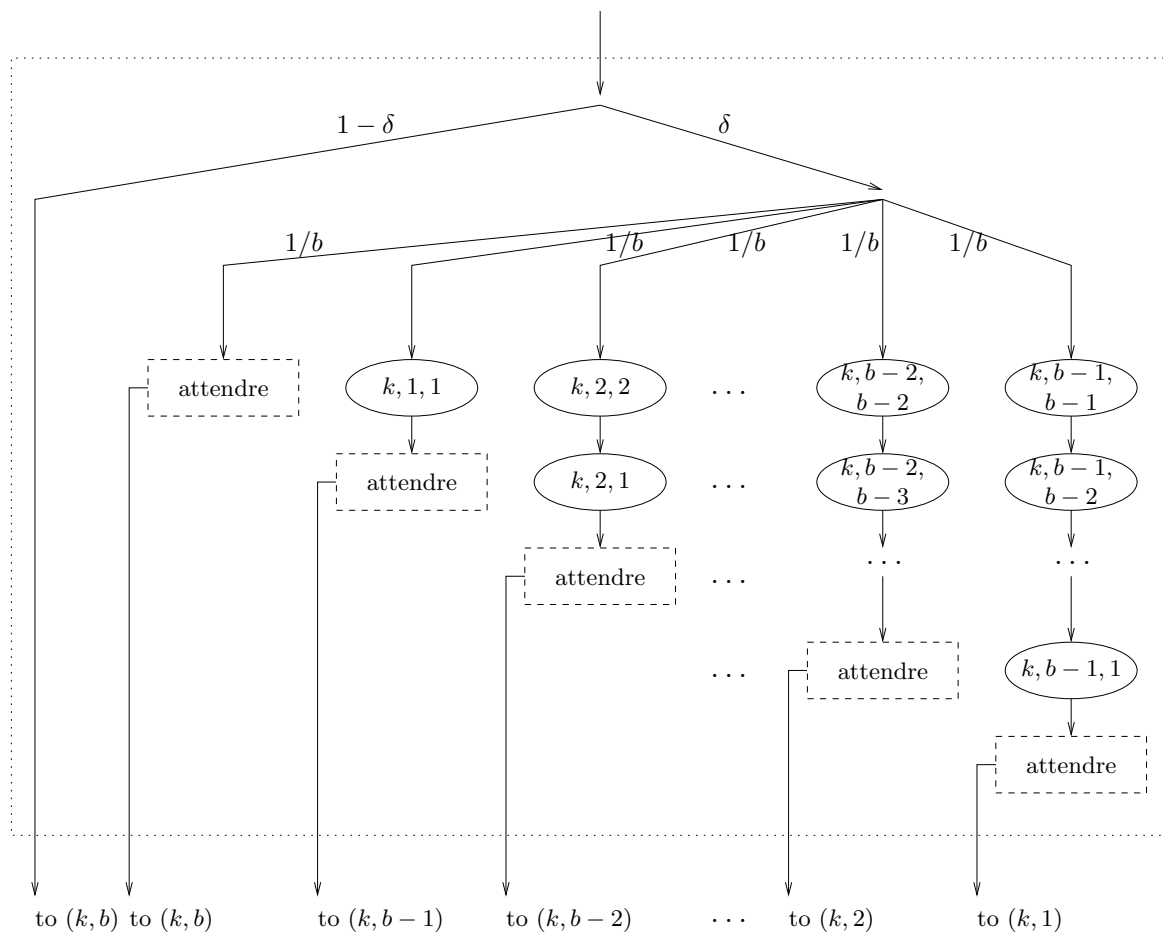


FIG. 4.15 – Report de *backoff* à la période d'activité suivante (boîte  $(k, \text{report}_b^B)$  de la figure 4.13).

correspond à la boîte (attendre). En début de la période d'activité suivante, le reste du *backoff* doit être consommé, ce qui correspond à l'état  $(k, b - x)$  de la figure 4.13.

La figure 4.16 décrit le contenu de la boîte "attendre". Nous considérons  $B$  comme étant le nombre de périodes de *backoff* qui sépare la fin de la période d'activité en cours et le début de la période d'activité suivante. Ceci est un délai de  $B$  périodes de *backoff*.

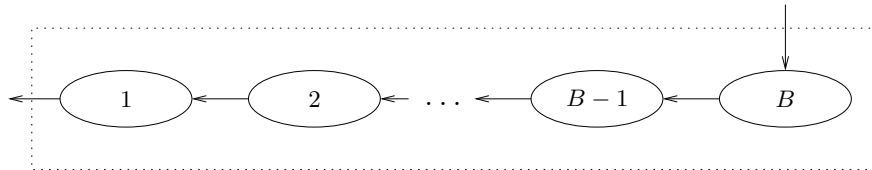


FIG. 4.16 – Attente entre la fin de la période d'activité en cours et la période d'activité suivante (boîte "attendre" des figures 4.14 et 4.15).

**Évaluation du délai engendré par les mécanismes de report** Dans notre équipe de recherche, ce point était une brique commune à deux sujets de thèse, ce travail a donc été mené en collaboration avec Mademoiselle Nassima Haddid. Nous avons évalué l'effet de ce report sur le délai d'accès au médium en utilisant la couche MAC de la norme IEEE 802.15.4<sup>4</sup>. Nous avons choisi un *scenario* simple constitué de deux nœuds : un CPAN et un coordinateur associé à ce CPAN. Le coordinateur envoie périodiquement 8 trames toutes les secondes pendant une durée de 250 secondes. Nous avons choisi un seul émetteur pour isoler le délai induit par le report de celui induit par le mécanisme de CSMA/CA slotté quand il y a plusieurs émetteurs concurrents. La taille des trames de 97 octets.

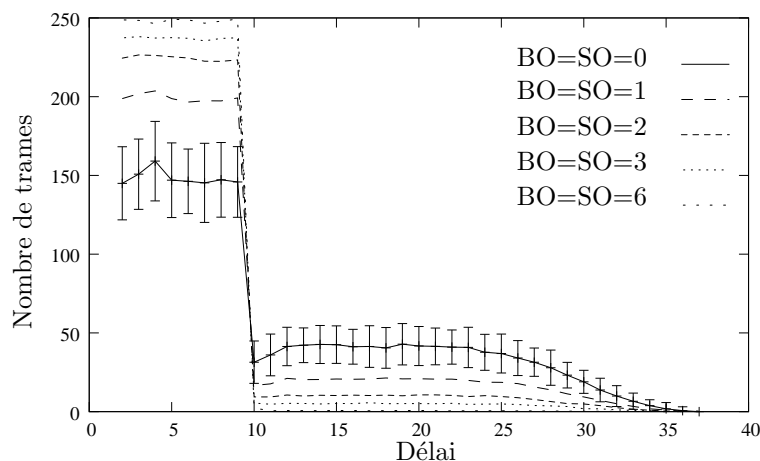


FIG. 4.17 – Distribution du délai d'accès au médium montrant l'effet du report.

La figure 4.17 montre la distribution du délai en terme de nombre de périodes de *backoff*. Nous avons considéré le cas d'un WPAN réactif pour lequel nous n'avons pas de périodes d'inactivité ( $BO = SO$ ).

<sup>4</sup>Ce travail a abouti à la publication [72].

Nous pouvons remarquer que malgré le fait qu'il n'y ait qu'un seul émetteur, des trames subissent un délai qui dépasse le délai maximum dû aux deux CCA et au tirage aléatoire du CSMA/CA slotté qui est de 9 périodes de *backoff*.

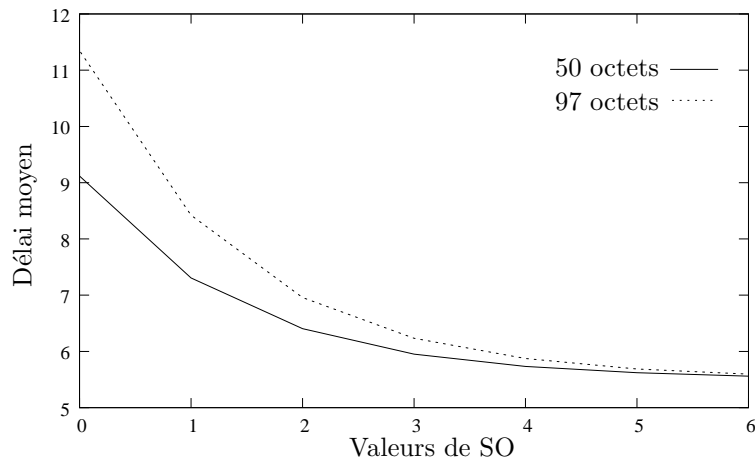


FIG. 4.18 – Effet de la taille des trames sur le délai induit par le report.

La figure 4.18 montre l'effet de la taille de trame sur le délai induit par le report. Nous pouvons constater que le délai augmente quand les trames sont plus grandes. Ceci est normal car la durée de la transaction est fonction de la taille des trames. Ainsi, les nœuds ayant une longue trame à envoyer subissent l'effet du report plus tôt avant la fin de la CAP que les nœuds avec des trames plus courtes.

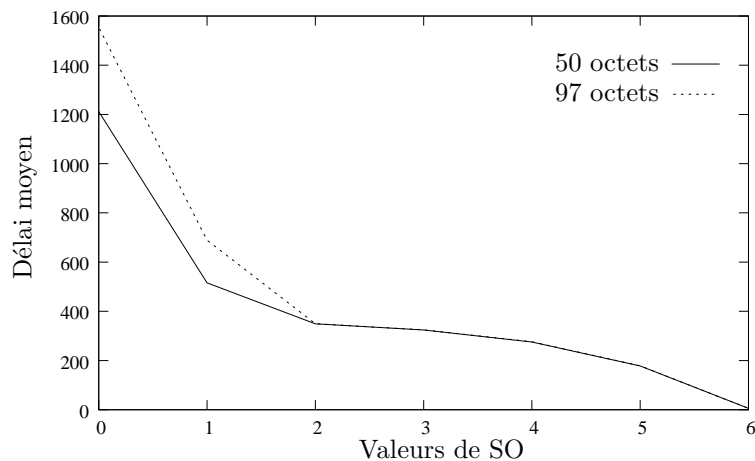


FIG. 4.19 – Délai induit par le report dans le cas d'un réseau économe en énergie.

La figure 4.19 montre que l'effet du report est encore plus significatif quand le réseau est économe en énergie (incluant des périodes d'inactivité :  $BO \neq SO$ ). Nous pouvons aussi constater que le report est plus significatif pour les petites valeurs de  $SO$  et que dans ce cas, son effet ne peut pas être ignoré. Ce phénomène sera constaté lors de nos mesures de délai pour MaCARI dans 3.1.5 et sera étudié dans le cadre d'une évaluation analytique du délai de bout-en-bout en 4.2.

### 3.1.3 Comparaison entre le CSMA/CA slotté et le *polling*

Le comportement du CSMA/CA slotté est comme tout autre algorithme d'accès au médium basé sur CSMA/CA, il ne garantit pas l'accès au canal ni la réception sans collision des données envoyées. En revanche, un mécanisme comme le *polling* où une station centrale interroge les entités à tour de rôle, offre un déterminisme d'accès au médium sans risque de collision mais n'est pas performant quand beaucoup d'entités sont inactives [73]. Dans la suite de cette partie nous allons comparer les résultats des performances du CSMA/CA slotté et d'un mécanisme de *polling* de type *round robin* pour lequel le coordinateur sollicite à tour de rôle toutes les feuilles qui lui sont associées pour récolter les données. Nous considérons uniquement le cas de la récolte de données, ce qui correspond à une étoile où toutes les feuilles sont des capteurs. Le but de cette comparaison est d'identifier à partir de quel pourcentage de feuilles inactives dans l'étoile le CSMA/CA slotté de la norme IEEE 802.15.4 devient plus performant que le mécanisme de *polling*.

Nous considérons différentes tailles d'étoiles, allant de 2 feuilles jusqu'à 24 feuilles. Toutes les feuilles actives ont une seule trame à envoyer. La période d'activité est déclenchée à la réception du beacon du coordinateur de l'étoile et les trames des feuilles sont à priori prêtes à l'envoi.

Les trames physiques font 50 octets de longueur. Les figures 4.20, 4.21, 4.22 et 4.23 représentent le temps nécessaire pour que toutes les feuilles aient envoyé chacune une trame au coordinateur et ceci pour un pourcentage de feuilles actives de 100 %, 75 %, 50 % et 25 % respectivement.

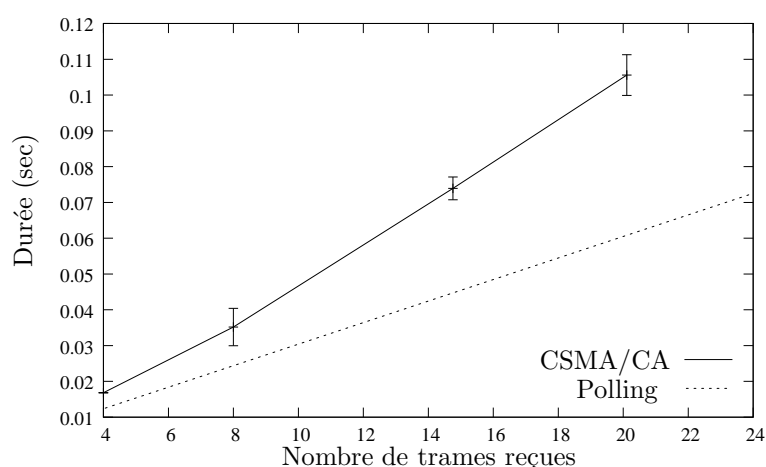


FIG. 4.20 – Temps de récolte avec 100 % des feuilles actives.

Le point A ( $x = 15,5$ ;  $y = 0,09$ ) de la figure 4.21 a été obtenu en considérant l'activité de 18 des 24 feuilles en CSMA/CA slotté. Seules 15,73 trames en moyenne sont reçues en un temps moyen de 0,09 secondes, 3 trames ont donc été perdues. Le point B ( $x = 18$ ;  $y = 0,066$ ) a été obtenu par la scrutation de toutes les feuilles, 18 ont répondu et 18 trames ont été reçues.

Les figures 4.20, 4.21, 4.22 et 4.23 montrent qu'avec un pourcentage entre 100 % et 75 % de feuilles

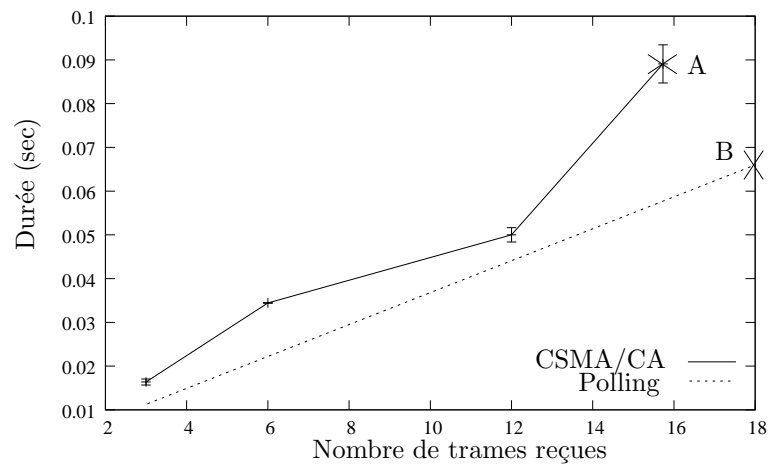


FIG. 4.21 – Temps de récolte avec 75 % des feuilles actives.

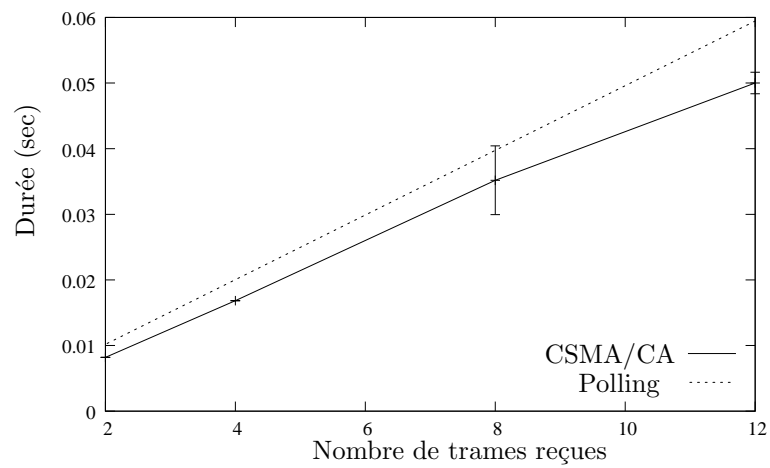


FIG. 4.22 – Temps de récolte avec 50 % des feuilles actives.

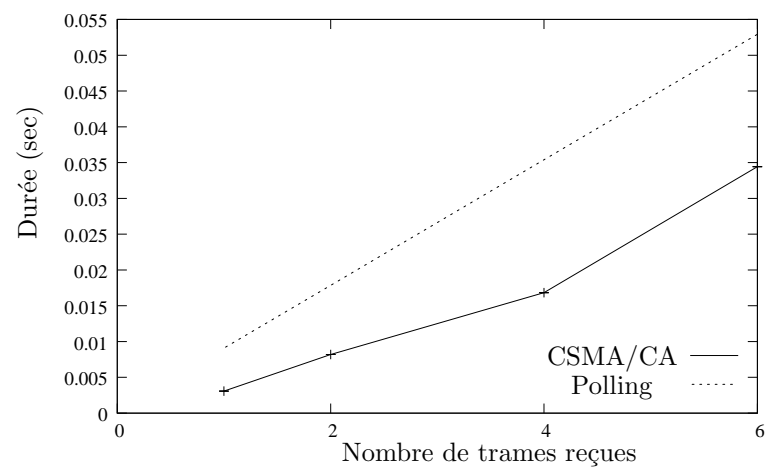


FIG. 4.23 – Temps de récolte avec 25 % des feuilles actives.

actives dans une étoile, le mécanisme de *polling* est plus performant en délai. Nous pouvons aussi constater comment l'augmentation du délai en fonction du nombre de trame reçues est exponentiel pour le CSMA/CA slotté. En revanche, quand le pourcentage de feuilles actives dans l'étoile est entre 75 % et 50 % les performances du CSMA/CA et du mécanisme de *polling* se rapprochent. Quand le pourcentage de feuilles actives dans l'étoile est en dessous de 50 % le CSMA/CA est plus efficace en terme de délai par rapport au *polling*. Cela est dû au fait que le *polling* perd du temps conséquent à interroger des feuilles inactives (qui n'ont pas de trames à envoyer).

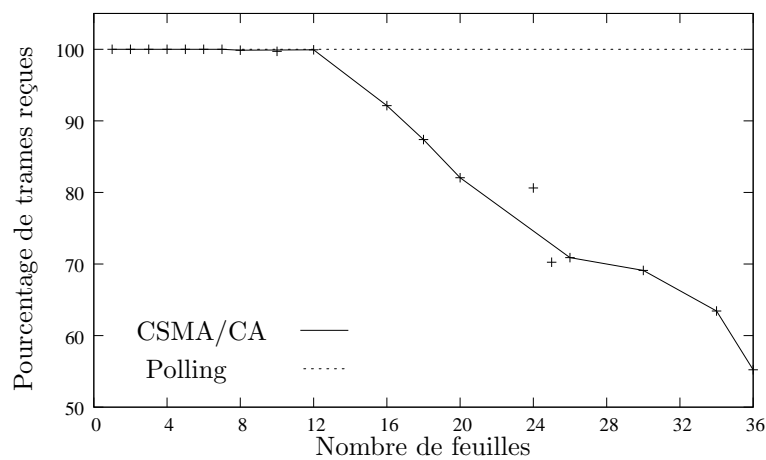


FIG. 4.24 – Performances de CSMA/CA slotté dans l'étoile.

D'après la courbe de la figure 4.24, il est intéressant de noter que le pourcentage de trames perdues est de 1 % entre 8 et 12 trames envoyées, alors qu'il dépasse les 12 % à partir de 18 trames envoyées. Ceci nous permet de conclure qu'avec nos hypothèses, le CSMA/CA slotté est un mécanisme fiable sous faible charge qui correspond à moins de 15 feuilles actives dans l'étoile avec chacune une trame à envoyer.

Dans le contexte du projet OCARI, nous supposons que le nombre maximum de feuilles actives dans une étoile ne dépasse pas 8 et que le nombre total de feuilles par étoile varie entre 12 et 16 feuilles. Ce qui nous permet de choisir le CSMA/CA slotté comme méthode d'accès pour gérer les activités des étoiles avec le mécanisme de GTS pour le trafic prioritaire pour lui assurer un degré plus élevé de déterminisme. Notons aussi que le CSMA/CA garde l'avantage de permettre à une nouvelle feuille de s'associer.

### 3.1.4 Amélioration des performances de la couche MAC par le découpage temporel

La segmentation temporelle du cycle global permet de séparer les trois activités suivantes : (i) la récolte d'informations produites dans les feuilles durant l'intervalle intra-étoile, (ii) le relais du trafic de type prioritaire durant l'intervalle relais garanti et (iii) le routage entre coordinateurs du trafic de type non-prioritaire durant  $[T_2; T_3]$ . Nous allons montrer que cette segmentation offre une meilleure utilisation du canal et augmente le débit du réseau.

Pour montrer l'amélioration des performances du réseau grâce à ce découpage temporel, nous allons nous baser sur les trois métriques suivantes : (j) le nombre global de collisions, (jj) le nombre global de trames reçues et (jjj) le débit utile. Cela dans 3 configurations différentes de MaCARI présentées sur la figure 4.25 :

1. Première configuration : une période de synchronisation suivie par une période  $[T_1; T_2]$  dans laquelle se passent toutes les communications de toutes les entités du réseau en même temps. Cette configuration se rapproche de la configuration temporelle de la solution *beacon-only period* décrite dans 2.5.1, dans ce cas toutes les étoiles ont la même durée d'activité qui est équivalente à la durée de  $[T_1; T_2]$ .
2. Deuxième configuration : une période de synchronisation suivie par une période  $[T_1; T_2]$  avec une période d'activité pour chaque étoile mais sans les intervalles de relais, suivi par une période  $[T_2; T_3]$ . Dans cette configuration, il n'y a pas de priorisation de trafic, tout le trafic est échangé entre les coordinateurs en CSMA/CA slotté dans  $[T_2; T_3]$ .
3. Troisième configuration : la configuration de MaCARI comme décrite dans la partie 1.1.1 du chapitre 3. Cette configuration supporte les deux types de trafic. Nous considérons que le trafic de type prioritaire représente 25 % du trafic total.

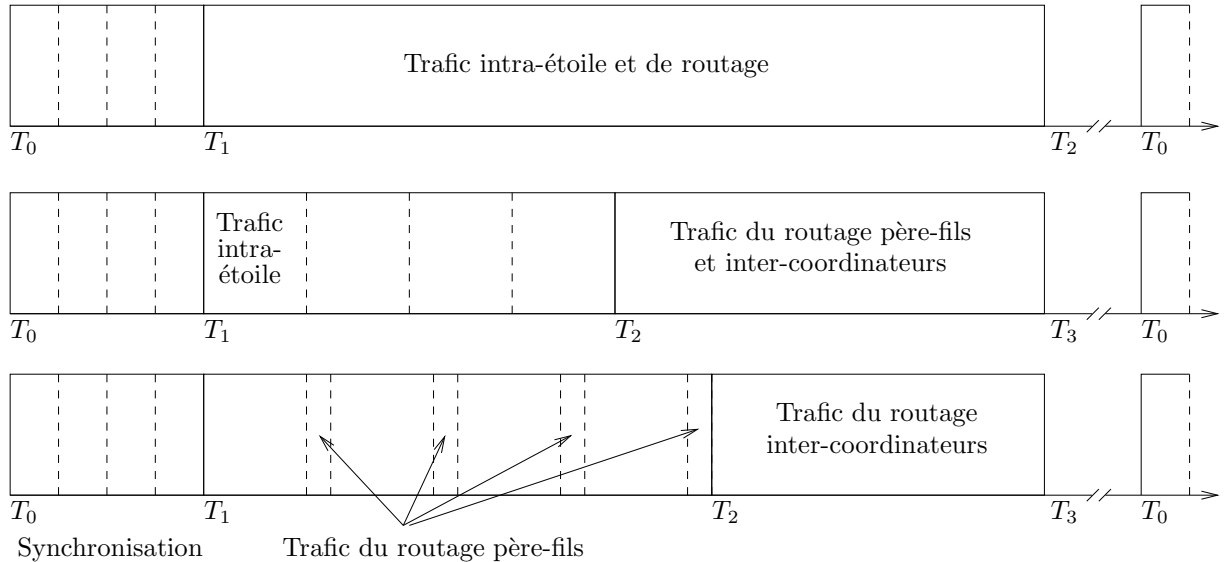


FIG. 4.25 – Les trois configurations de MaCARI.

Pour que les comparaisons soient légitimes, la durée du cycle global et de la partie active sont les mêmes pour toutes les configurations. Ainsi, dans la deuxième configuration, la durée de  $[T_1; T_3]$  est égale à la durée de  $[T_1; T_2]$  de la première configuration. Pour ajouter les intervalles de relais dans la troisième configuration, nous avons ajouté un intervalle de temps supplémentaire pour chaque période d'activité,



et nous avons réduit la durée de  $[T_2; T_3]$  de la somme de tous les intervalles de relais ajoutés. Cela nous permet de comparer les performances des trois configurations tout en considérant le même découpage espace temporel appliqué pour le routage et la récolte (les intervalles de relais +  $[T_2; T_3]$  de la troisième configuration =  $[T_2; T_3]$  de la deuxième configuration, le temps consacré au routage est un invariant dans les deux dernières configurations.).

Nous avons considéré, pour chacune des trois configurations, 5 *scenarii* de test :

1. 9 coordinateurs et 25 feuilles
2. 9 coordinateurs et 36 feuilles
3. 16 coordinateurs et 49 feuilles
4. 16 coordinateurs et 64 feuilles
5. 25 coordinateurs et 81 feuilles

Les topologies sont construites avec les paramètres  $Lm = 5$ ,  $Rm = 3$  et  $Cm = 8$ . Nous avons considéré un trafic périodique généré uniquement par les feuilles toutes les secondes. Chaque feuille envoie 16 trames. Chaque trame fait 22 octets de longueur.

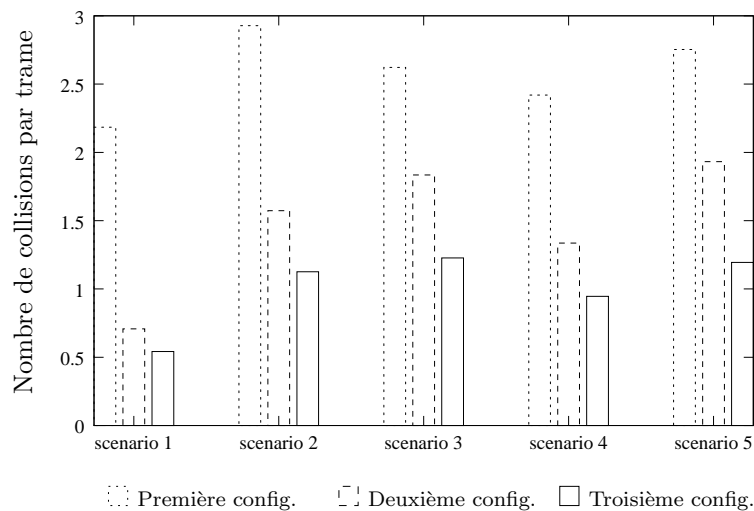


FIG. 4.26 – Nombre de collisions par trame envoyée selon la configuration pour chacun des 5 *scenarii* pour la même quantité de trafic.

La figure 4.26 montre le gain en évitement de collision quand nous appliquons une segmentation temporelle et nous activons les nœuds par groupes dans la deuxième configuration, au lieu de les activer tous en même temps comme c'est le cas dans la première configuration. Un gain supplémentaire est obtenu quand nous garantissons l'accès au canal pour le trafic prioritaire dans la troisième configuration.

Les figures 4.27 et 4.28 nous permettent de constater que le pourcentage de trames reçues augmente dans la deuxième et surtout troisième configurations par rapport à celui de la première configuration, de

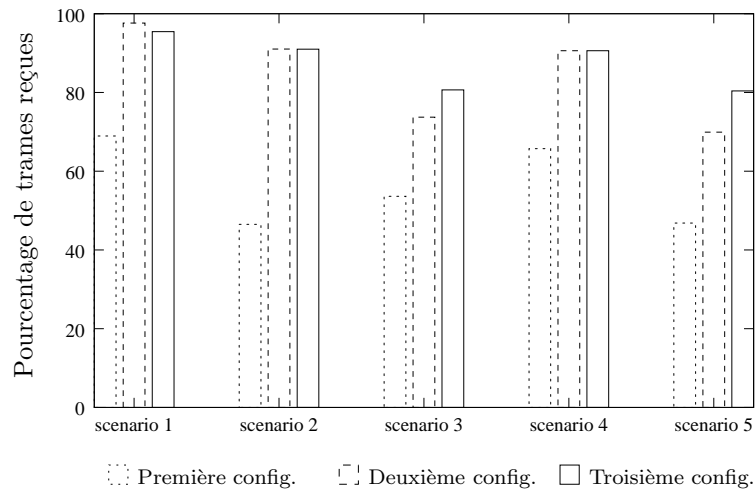


FIG. 4.27 – Pourcentage de trames reçues selon la configuration pour chacun des 5 *scenarii* pour la même quantité de trafic.

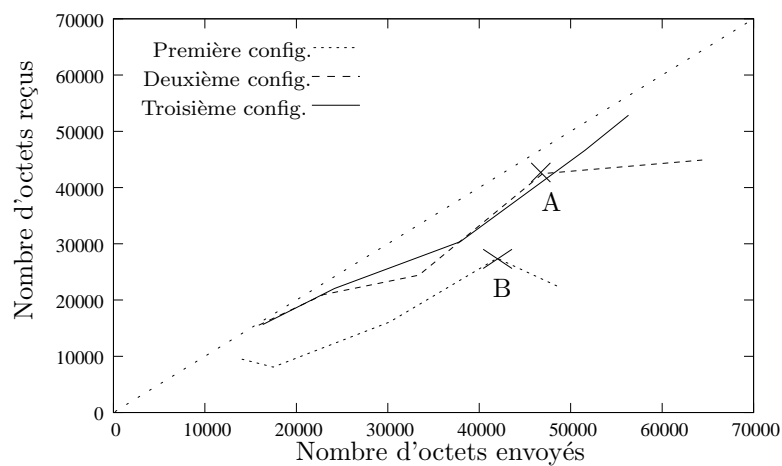


FIG. 4.28 – Nombre d'octets reçus par rapport au nombre d'octets envoyés selon la configuration pour chacun des 5 *scenarii*.

même pour le nombre d'octets reçus. Prenons l'exemple des points A et B de la figure 4.28. Ces deux points correspondent au *scenario* numéro 4 avec la même topologie du réseau (16 coordinateurs et 64 feuilles). Pour cette configuration, en appliquant la première configuration, nous avons eu en moyenne sur 100 répliques, 41956.2 octets envoyés dont 27438.62 octets acquittés. En appliquant la deuxième configuration, nous avons eu en moyenne 46946.68 octets envoyés dont 42474.96 octets ont été acquittés. Cela prouve que le CSMA/CA slotté de la norme IEEE 802.15.4 est plus performant quand nous activons séquentiellement les nœuds par groupe (par étoile dans notre cas).

La segmentation temporelle apporte une amélioration radicale à la bande passante offerte par le réseau, il reste à étudier les incidences sur le temps de transit.

### 3.1.5 Délai de bout-en-bout

Dans cette partie, nous avons considéré la topologie représentée sur la figure 4.29. Nous souhaitons mesurer le délai de bout-en-bout d'un trafic généré par les feuilles 9 et 12. Dans un premier temps, nous avons généré 100 trames par chacune des feuilles de type trafic non-prioritaire à destination du CPAN avec une périodicité d'une trame toutes les 1,5 secondes. Ensuite, nous avons généré 100 trames de type trafic prioritaire par les feuilles 9 et 12, et 100 trames de type trafic non-prioritaire par chacune des autres feuilles. Le but est de protéger l'accès au médium pour le trafic généré par les feuilles 9 et 12. La fréquence de génération de trames est d'une trame par seconde.

Les trames de type prioritaire sont envoyées par les feuilles au coordinateur de leur étoile durant les intervalles de temps de GTS. Les autres trames sont envoyées en CSMA/CA slotté durant l'intervalle intra-étoile.

Les choix des intervalles temporels ont été faits de façon à ce qu'ils puissent être retenus dans la phase de maquettage sachant que le cadencement préconisé par la norme IEEE 802.15.4 est un multiple de 320  $\mu$ s. Dans cette configuration, nous avons choisi un temps de récolte de 0,123 s. Ainsi, la durée de  $[T_1; T_2]$  sera de  $6 * 0,123$  s (dans cet exemple nous avons choisi de ne pas donner de période d'activité pour le CPAN). Avec  $[T_2; T_3] = [T_1; T_2]$ ,  $[T_0; T_1] = 0,072$  s et un cycle global<sup>5</sup> de 1,877 s.

Les courbes des figures 4.30 et 4.31 représentent le délai de bout-en-bout des 100 trames générées par les feuilles 9 et 12 respectivement et à destination du nœud O de la topologie affichée sur la figure 4.29. Les résultats nous permettent de constater que 100 % des trames ont été reçues par le nœud 0. En revanche, le délai de bout-en-bout dépasse des fois la durée d'un cycle global. Cela est dû au report des trames tardives qui arrivent à la couche MaCARI vers la fin de l'intervalle de temps intra-étoile pour le trafic non-prioritaire et vers la fin de l'intervalle dédié aux GTS pour le trafic prioritaire.

---

<sup>5</sup>1,877 seconde correspond à peu près à un  $BO = 7$ .

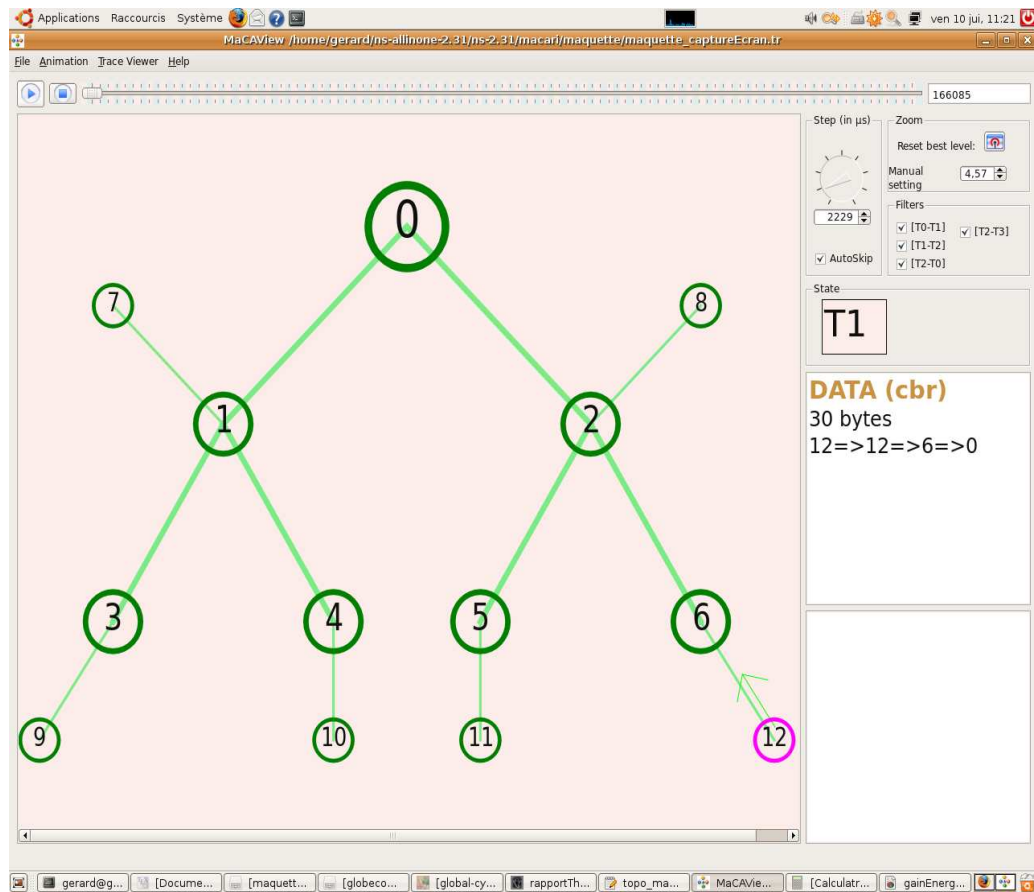


FIG. 4.29 – Capture d’écran de notre animateur de simulation montrant une topologie de 6 coordinateurs avec 1 feuille chacun et 1 CPAN.

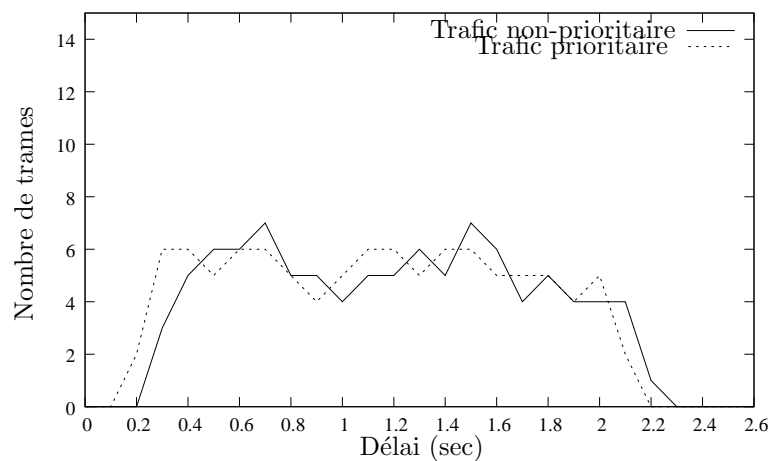


FIG. 4.30 – Simulation : Distribution des délais de bout-en-bout pour la feuille 9 de la topologie 4.29.

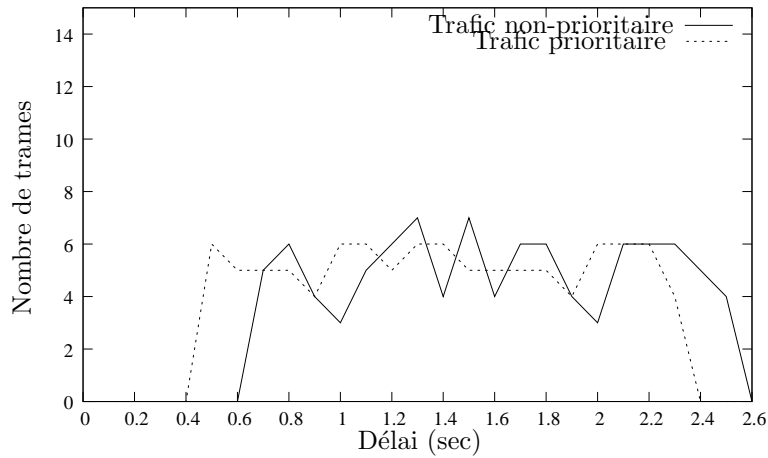


FIG. 4.31 – Simulation : Distribution des délais de bout-en-bout pour la feuille 12 de la topologie 4.29.

## 3.2 Maquettage

Dans cette partie, nous allons reprendre les *scenarii* de simulation et les réaliser sur les cartes B2400ZB–tiny pour prouver la faisabilité du découpage temporel et pour faire le rapprochement avec les résultats de simulations.

### 3.2.1 Évaluation de la durée des périodes d’activité

Pour faire le rapprochement entre la simulation et les mesures réelles, dans le cadre d’une collaboration avec le LATTIS<sup>6</sup> qui a abouti à une publication commune [74], nous avons construit les mêmes *scenarii* décrits dans 3.1.1 avec les cartes B2400ZB–tiny.

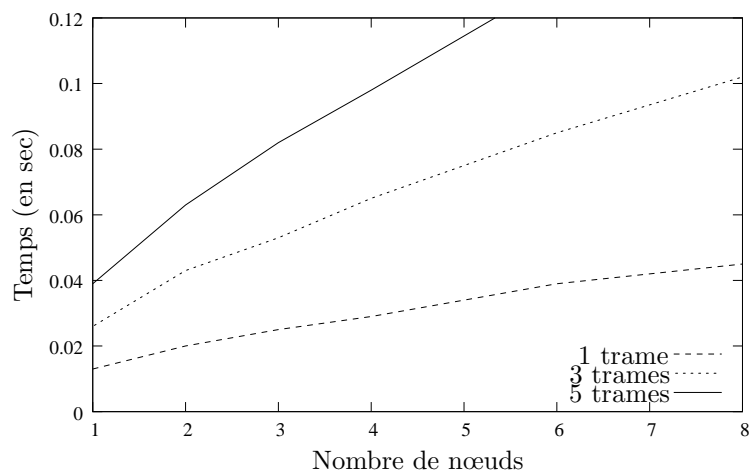


FIG. 4.32 – Mesures : Temps nécessaire pour faire écouler une certaine quantité de trafic dans une étoile en fonction du nombre de feuilles et du nombre de trames générées par feuille.

Nous pouvons constater que les résultats des figures 4.32 et 4.10 sont assez similaires, de même pour

<sup>6</sup>Les résultats des mesures des figures 4.32 et 4.33 ont été produits par notre partenaire LATTIS dans le cadre d’une publication commune.

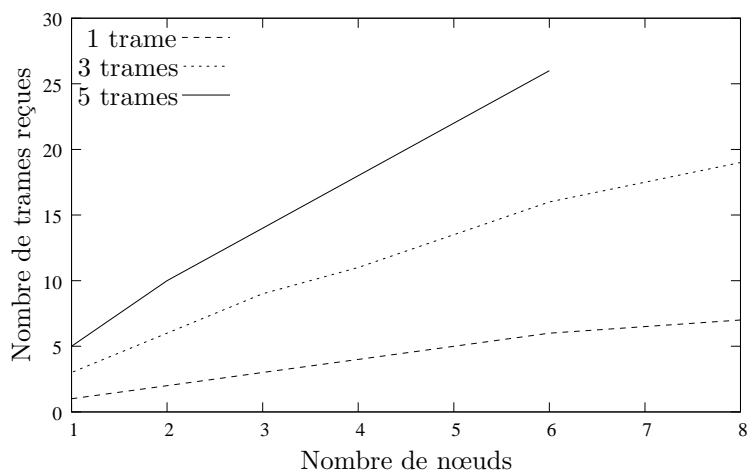


FIG. 4.33 – Mesures : Nombre de trames reçues par le coordinateur d'une étoile en fonction du nombre de feuilles et du trafic généré par chaque feuille.

les figures 4.33 et 4.11. L'évolution des courbes est la même avec un léger décalage qui est dû au fait que le temps de traitement du microcontrôleur des cartes n'est pas pris en compte sur le simulateur alors que ce temps induit des retards dans la réactivité et le fonctionnement des cartes.

### 3.2.2 Délai de bout-en-bout

Avec les mêmes paramètres que dans 3.1.5, nous avons repris les mesures de délai de bout-en-bout pour la configuration de la figure 4.29 reconstruite avec les cartes B2400ZB-tiny.

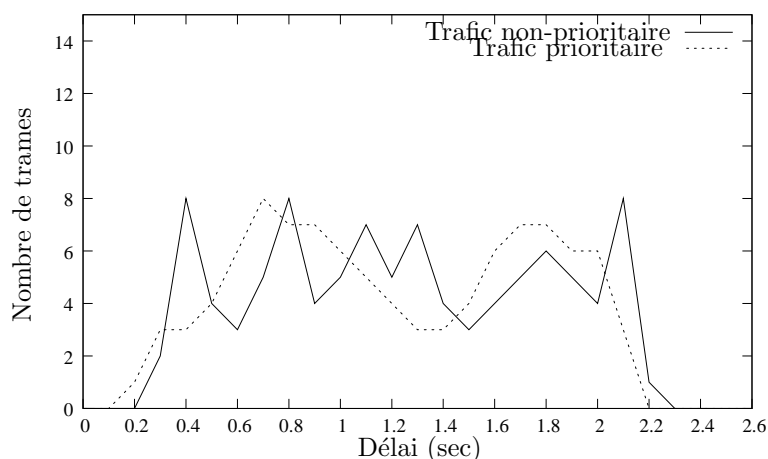


FIG. 4.34 – Mesures : Distribution des délais de bout-en-bout pour la feuille 9 de la topologie 4.29.

Nous pouvons remarquer que les résultats se rapprochent et les distributions de délai ont la même allure, et s'étalent sur le même intervalle ([0,1 et 0,2; 2,2 et 2,3] pour la feuille 9, et [0,4 et 0,6; 2,4 et 2,6] pour la feuille 12). Nous retrouvons le même effet de report observé sur les figures 4.30 et 4.31. Cela prouve que notre implémentation et notre modélisation sur le simulateur, des différents aspects détaillés dans 1.1 du chapitre 4 sont réalistes.

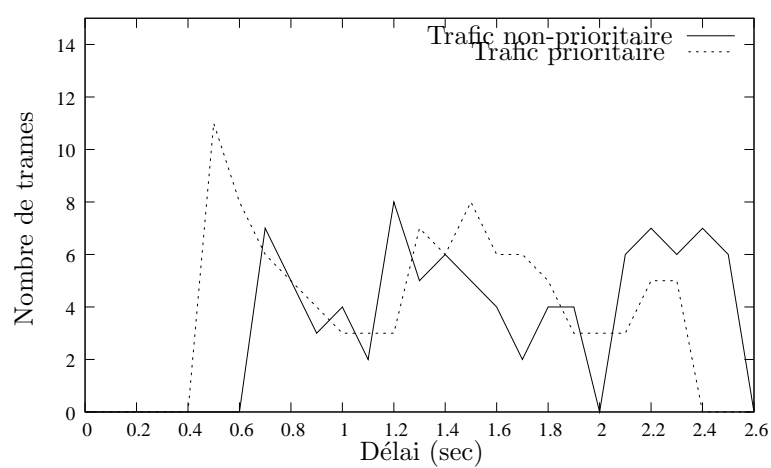


FIG. 4.35 – Mesures : Distribution des délais de bout-en-bout pour la feuille 12 de la topologie 4.29.

## 4 Étude analytique

Dans cette partie nous terminons le chapitre résultats par une étude analytique de l'économie d'énergie et du délai de bout-en-bout offerts par MaCARI.

### 4.1 Étude et évaluation de l'économie d'énergie

Dans cette partie nous allons quantifier la durée des deux intervalles  $[T_0; T_1]$  et  $[T_1; T_2]$  qui dépendent du nombre d'entités dans le réseau et du trafic généré. Pour ce faire, nous allons nous baser sur les résultats du maquetage donnés sur la figure 4.32.

#### 4.1.1 Calcul de la durée de $[T_0; T_1]$

La durée de  $[T_0; T_1]$  dépend de deux paramètres : (i) le nombre de coordinateurs et (ii) la taille du *beacon*. Plus la taille du *beacon* augmente plus le temps de traitement et d'envoi augmente. Suite à des mesures effectuées sur les cartes, nous avons constaté que le temps nécessaire pour traiter et envoyer un *beacon* de taille minimum (quand le CPAN est tout seul dans le réseau) est de  $8 \times 320 \mu s$ . Ensuite, pour chaque adresse supplémentaire (ou deux octets supplémentaires suite à une nouvelle association), le temps augmente d'une durée de l'ordre de  $320 \mu s$ .

Cela nous donne la formule suivante pour calculer la durée de  $[T_0; T_1]$  en fonction du nombre de coordinateurs  $n$  :

$$[T_0; T_1] = n * (8 + n * 0,32) \text{ ms.}$$

#### 4.1.2 Calcul de la durée de $[T_1; T_2]$

La durée de  $[T_1; T_2]$  dépend essentiellement de la durée des périodes d'activité allouées aux différentes étoiles du réseau. Chaque période d'activité est découpée en deux intervalles : (i) un intervalle de temps pour gérer l'activité intra-étoile et (ii) un intervalle de temps, qui est l'intervalle de relais garanti, pour échanger entre le coordinateur de l'étoile et son père. La durée de l'intervalle intra-étoile dépend de la charge générée par l'ensemble des feuilles de l'étoile, et la durée de l'intervalle de relais dépend de la charge du trafic de type prioritaire.

$$[T_1; T_2] = n * (duree\_intra\_etoile + duree\_intervalle\_relais)$$

#### 4.1.3 Tableau récapitulatif des durées de $[T_0; T_1]$ et $[T_1; T_2]$

Nous supposons que pour un réseau donné, le nombre maximum de feuilles actives par étoile est le même pour toutes les étoiles du réseau et que la durée de l'intervalle intra-étoile doit être suffisante



pour que chaque feuille de l'étoile puisse envoyer 1 trame de 50 octets (niveau physique) avant la fin de l'intervalle de collecte.

Nous avons considéré 4 tailles d'étoiles différentes : 2, 4, 6 et 8 feuilles actives maximum par étoile. Ce qui correspond aux 4 durées d'intervalle intra-étoile suivantes : 20, 30, 45 et 50 ms respectivement (issues de 4.32). Pour les intervalles de relais, nous considérons 20 ms qui permettent d'échanger 10 trames de 50 octets sans CSMA/CA mais avec un protocole point à point étudié en 3.1.3 du chapitre 3. Nous n'avons pas considéré ici l'augmentation de débit qui pourrait résulter de l'agrégation de ces trames. Ce point sera repris dans les perspectives de ce travail.

Le tableau 4.2 récapitule les durées de  $[T_0; T_1]$  et  $[T_1; T_2]$  en fonction du nombre de coordinateurs et le nombre de feuilles actives maximum par coordinateur.

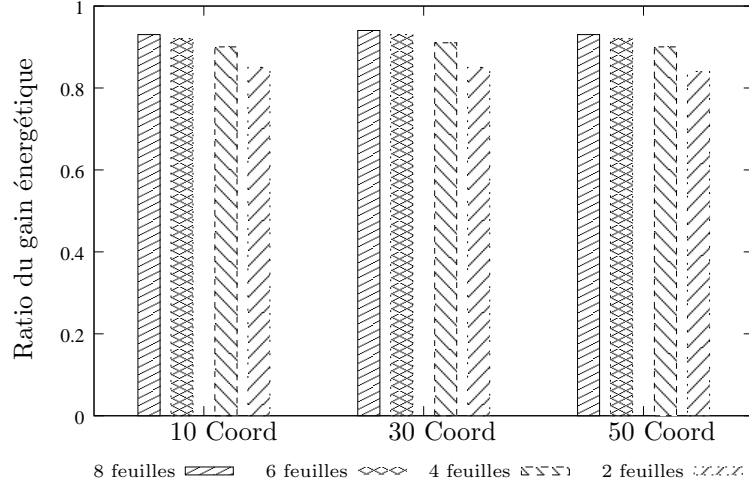
Nombre de Coord.	Nombre de feuilles actives par Coord.	Durée intra-étoile (sec)	Durée de $[T_0; T_1]$ (sec)	Durée de $[T_1; T_2]$ (sec)
50	8	0.05	1,2	3,5
50	6	0.045	1,2	3,25
50	4	0.03	1,2	2,5
50	2	0.02	1,2	1,5
30	8	0.05	0,528	2,1
30	6	0.045	0,528	1,95
30	4	0.03	0,528	1,5
30	2	0.02	0,528	0,9
10	8	0.05	0,112	0,7
10	6	0.045	0,112	0,65
10	4	0.03	0,112	0,5
10	2	0.02	0,112	0,3

TAB. 4.2 – Taille des périodes  $[T_0; T_1]$  et  $[T_1; T_2]$  en fonction du nombre de nœuds.

Nous avons choisi de prendre  $[T_2; T_3] = [T_1; T_2]$  et  $[T_3; T_0] = [T_1; T_3]$  dans un premier temps (la durée de  $[T_3; T_0]$  dépend de la réactivité souhaitée par l'application, plus elle est petite plus le réseau est réactif mais peu économe en énergie, plus la durée de  $[T_3; T_0]$  est grande plus le réseau est économe en énergie mais moins réactif).

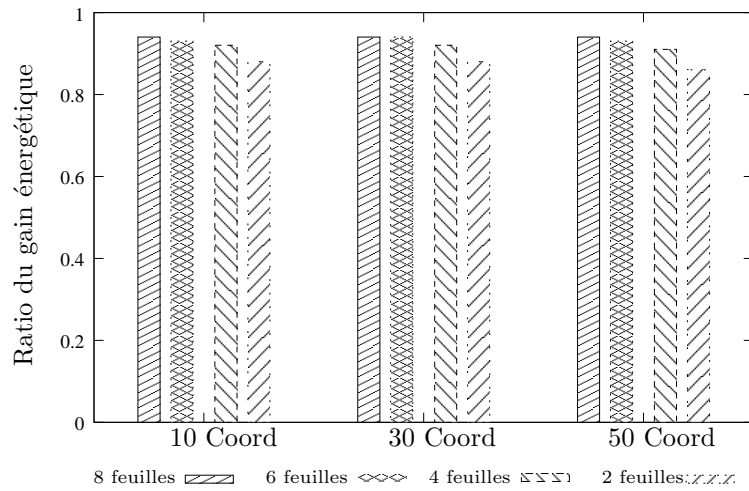
La figure 4.36 montre le gain d'énergie obtenu en fonction de la taille du réseau en faisant varier le nombre de coordinateurs et le nombre de feuilles par coordinateur. Ce gain est calculé en faisant le ratio de la durée passée en mode sommeil par rapport à une configuration où les entités sont en mode éveillé en permanence.

Nous atteignons les 94 % d'économie d'énergie pour les configurations avec 8 feuilles par coordinateur. Notons qu'avec plus de feuilles nous trouvons un gain d'énergie plus grand car les feuilles consomment


 FIG. 4.36 – Gain énergétique avec  $[T_2; T_3] = [T_1; T_2]$  et  $[T_3; T_0] = [T_1; T_2] + [T_2; T_3]$ .

moins d'énergie que les coordinateurs dans un cycle global, alors que le nombre de coordinateurs n'a que peu d'effet sur le gain énergétique.

Pour étudier l'effet de la durée de  $[T_2; T_3]$  sur l'économie d'énergie, nous avons repris nos calculs mais cette fois-ci avec  $[T_2; T_3] = [T_1; T_2]/2$ . Ceci pénalise le trafic non-prioritaire en lui donnant moins de temps pour qu'il soit routé. La figure 4.37 montre que nous atteignons un gain entre 1 et 3 % supplémentaire par rapport aux résultats de la figure 4.36.


 FIG. 4.37 – Gain énergétique avec  $[T_2; T_3] = [T_1; T_2]/2$  et  $[T_3; T_0] = [T_1; T_2] + [T_2; T_3]$ .

Pour mettre en relief le gain énergétique offert par MaCARI, nous allons comparer ce gain à celui offert par la première configuration considérée dans 3.1.4 de la partie 3. La figure 4.38 montre le gain énergétique offert par ces deux solutions en fonction de la durée de  $[T_3; T_0]$  et le nombre de coordinateurs. Nous avons choisi 4 valeurs différentes pour la durée de  $[T_3; T_0]$  : 0 ms,  $D_{T_1 T_3}/2$ ,  $D_{T_1 T_3}$  et  $D_{T_1 T_3} * 2$ .

Nous pouvons noter que MaCARI réalise une économie d'énergie qui dépasse les 85 % même quand

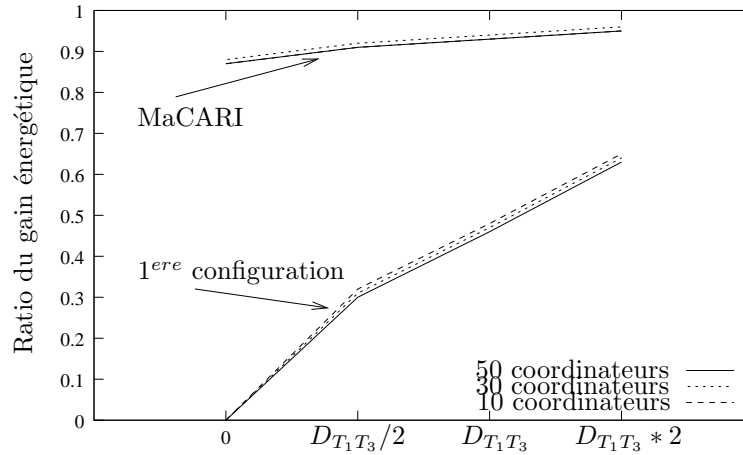


FIG. 4.38 – Abaque : gain énergétique en fonction de la durée de  $[T_3; T_0]$  pour une configuration avec 8 feuilles par coordinateur.

$[T_3; T_0] = 0$ , et reste globalement nettement meilleur pour l'ensemble des cas étudiés.

#### 4.1.4 Application numérique pour les *scenarii* de la partie 3.1.4

Nous revenons sur les 5 *scenarii* définis dans 3.1.4 pour estimer le gain énergétique en fonction de la durée de  $[T_3; T_0]$ . Nous considérons uniquement la première et la troisième configuration temporelle.

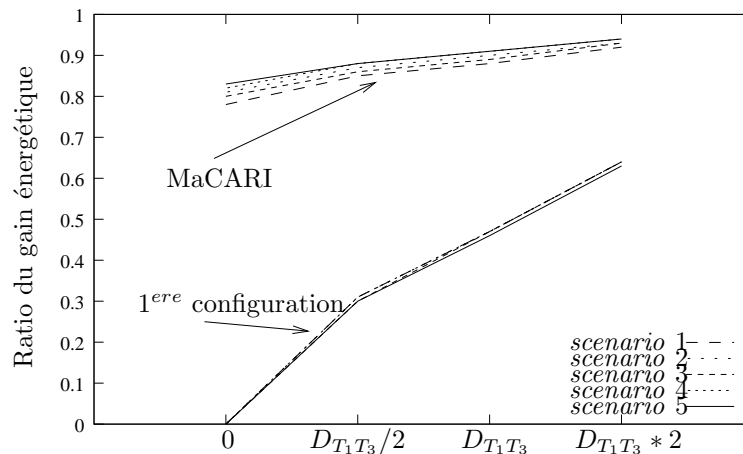


FIG. 4.39 – Abaque : Gain énergétique pour les 5 *scenarii*.

Notons comment le nombre de feuilles n'a pas effet sur le gain énergétique dans la première configuration car les feuilles et les coordinateurs suivent le même rythme d'activité, ce qui explique la superposition des courbes de l'abaque inférieure. Néanmoins, nous remarquons que les courbes de l'abaque supérieure, qui correspondent à la troisième configuration temporelle, sont disjointes donc légèrement sensibles au nombre de feuilles par étoile.

## 4.2 Étude et évaluation du délai de bout-en-bout

Dans cette partie, nous allons donner la taille d'une configuration du réseau qui garantit un délai de bout-en-bout inférieur à un certain seuil pour les trames de type prioritaire (ceci pour répondre à l'objectif de déterminisme qui devait caractériser MaCARI). Nous allons nous baser sur les valeurs calculées à partir des mesures effectuées en maquettage pour considérer les durées des différents intervalles du cycle global.

En effet, nous allons nous intéresser au pire des cas pour le calcul du délai. Ce pire des cas correspond à une trame générée par une feuille et reçue par la couche MaCARI de cette feuille vers la fin de son GTS. Ainsi, cette trame est reportée pour être envoyée dans le GTS du cycle global suivant. La figure 4.40 illustre l'instant de réception, par la couche MaCARI d'une feuille, d'une trame générée par les couches supérieures.

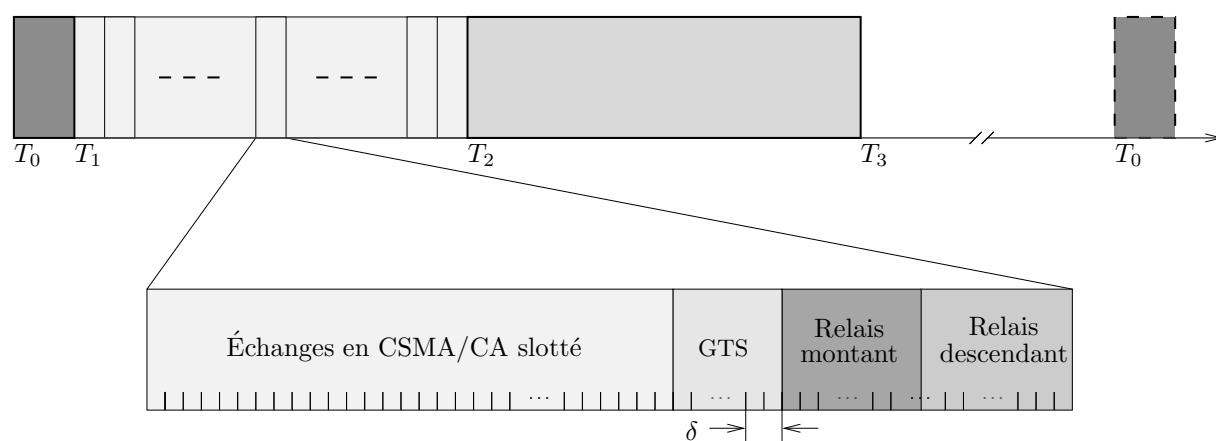


FIG. 4.40 – Définition du pire des cas pour l'évaluation du délai de bout-en-bout pour un trafic prioritaire.

$\delta$  est un temps à partir de la fin du GTS de la feuille qui a une trame de type prioritaire à émettre. Ce temps définit un seuil qui implique qu'une requête d'émission reçue avant la fin du GTS ne pourrait être traitée qu'au cycle global suivant. Nous supposons que le dimensionnement des périodes d'activité est fait en fonction du trafic généré, et que la durée d'une période d'activité est suffisante pour envoyer tout le trafic généré dans les feuilles d'une étoile. Ainsi, les trames reportées sont uniquement celles qui sont générées par les couches supérieures.

Nous supposons aussi que la durée des intervalles de relais est suffisante, ainsi, les couples de coordinateurs père-fils ont le temps pour échanger tout le trafic de type prioritaire en attente en appliquant les adaptations proposées dans 3.1.3 et 3.4.1 du chapitre 3 sans augmenter la taille des périodes d'activité dans  $[T_1; T_2]$ .

#### 4.2.1 Trafic prioritaire montant à destination du CPAN

Nous supposons dans ce cas que le trafic est généré dans les feuilles à destination du CPAN, et que le séquençement des périodes d'activité dans  $[T_1; T_2]$  favorise le flux montant. Nous considérons le pire des cas correspondant au cas particulier :

- la trame est générée par une feuille qui appartient à l'étoile dont la période d'activité est la première dans  $[T_1; T_2]$ ,
- le dernier saut en relais garanti est effectué durant le dernier intervalle de relais de  $[T_1; T_2]$ .

Nous supposons aussi que  $[T_1; T_2]$  est découpé équitablement entre les étoiles.

La trame générée en fin de la première période d'activité subit un report, elle est donc envoyée en début de cette période du cycle global suivant, ce qui fait une attente d'accès au médium d'un cycle global moins la durée d'une période d'activité. Ensuite, cette trame est relayée durant l'intervalle de relais de la première période d'activité et arrive à destination finale durant l'intervalle de relais de la dernière période d'activité (car il s'agit du CPAN).

Cela nous donne le délai de bout-en-bout, approximé à la valeur maximum, suivant :

$$delai\_bout\_en\_bout \leq duree\_cycle\_global + [T_1; T_2]$$

Pour avoir un délai de bout-en-bout inférieur à 1 seconde, il suffit d'avoir :

$$duree\_cycle\_global + [T_1; T_2] \leq 1seconde \text{ donc}$$

$$[T_0; T_1] + [T_1; T_2] + [T_2; T_3] + [T_3; T_0] + [T_1; T_2] \leq 1 \quad (4.1)$$

Soit  $n$  le nombre de coordinateurs, alors d'après les valeurs données dans 4.1 :

$$[T_0; T_1] = n * (0,008 + n * 0,00032) = 0,00032 * n^2 + 0,008 * n$$

$[T_1; T_2] = n * (0,05 + 0,01) = 0,06 * n$ , pour une topologie avec 8 feuilles actives maximum par coordinateur 50 *ms* sont nécessaire pour la récolte et 10 *ms* pour la durée de l'intervalle de relais garanti montant.

En prenant pour  $[T_2; T_3]$  la même durée que  $[T_1; T_2]$ , nous obtenons l'équation 4.1 en terme de  $n$  :

$$0,00032 * n^2 + 0,188 * n + [T_3; T_0]$$

En supposant que la durée de  $[T_3; T_0]$  peut être ramenée à 0, nous obtenons une équation du second degré avec le nombre de coordinateurs comme inconnue :  $0,00032 * n^2 + 0,188 * n - 1 = 0$ . La solution nous donne deux valeurs de  $n$ , dont une est négative à rejeter et l'autre est 5,3. Cela nous donne une topologie maximale avec 5 coordinateurs et 8 feuilles actives par coordinateur pour garantir un délai de bout-en-bout inférieur à 1 seconde.

Pour une topologie de 6 feuilles par coordinateur, nous avons  $[T_1; T_2] = n * (0,045 + 0,01) = n * 0,055$ , ce qui nous donne l'équation suivante :  $0,00032 * n^2 + 0,173 * n - 1 = 0$  qui a comme solution positive 5,7. Cela nous donne une topologie maximale de 5 coordinateurs et 6 feuilles par coordinateur pour garantir un délai de bout-en-bout inférieur à 1 seconde.

Pour une topologie de 4 feuilles actives par coordinateur, nous avons  $[T_1; T_2] = n * (0,03 + 0,01) = n * 0,04$ , ce qui nous donne l'équation suivante :  $0,00032 * n^2 + 0,128 * n - 1 = 0$  qui a comme solution positive 7,8. Cela nous donne une topologie maximale de 7 coordinateurs et 4 feuilles actives par coordinateur pour garantir un délai de bout-en-bout inférieur à 1 seconde.

Pour une topologie de 2 feuilles actives par coordinateur, nous avons  $[T_1; T_2] = n * (0,02 + 0,01) = n * 0,03$ , ce qui nous donne l'équation suivante :  $0,00032 * n^2 + 0,098 * n - 1 = 0$  qui a comme solution positive 9,4. Cela nous donne une topologie maximale de 9 coordinateurs et 2 feuilles actives par coordinateur pour garantir un délai de bout-en-bout inférieur à 1 seconde.

De la même manière, pour maximiser le délai de bout-en-bout à 2 secondes, nous obtenons les équations suivantes pour 8, 6, 4 et 2 feuilles actives maximum respectivement :  $0,00032 * n^2 + 0,188 * n - 2 = 0$ ,  $0,00032 * n^2 + 0,173 * n - 2 = 0$ ,  $0,00032 * n^2 + 0,128 * n - 2 = 0$  et  $0,00032 * n^2 + 0,098 * n - 2 = 0$ . Cela nous donne 10,4, 11,25, 15 et 19,2 comme solutions positives respectivement.

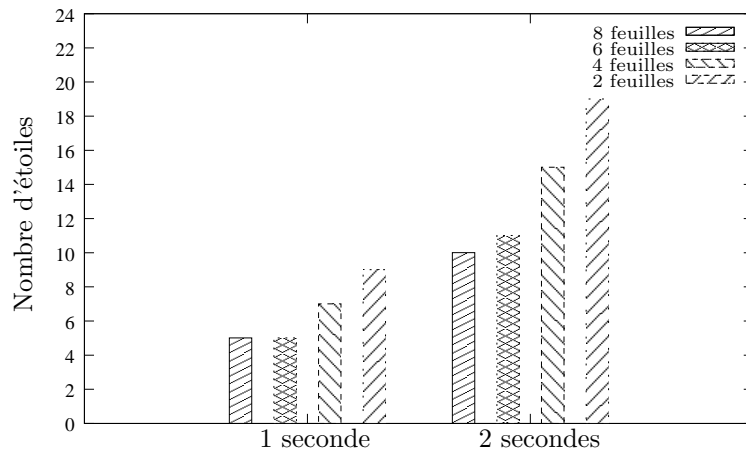


FIG. 4.41 – Taille du réseau avec  $[T_2; T_3] = [T_1; T_2]$  et un trafic à destination du CPAN.

La figure 4.41 montre qu'il est possible de satisfaire un majorant du temps de transit de bout-en-bout en agissant sur la taille de la configuration (nombre de coordinateurs et nombre de feuilles). Ainsi, pour garantir un délai de bout-en-bout qui ne dépasse pas la seconde, la taille du réseau ne peut pas dépasser 5 étoiles ayant entre 6 et 8 feuilles actives chacune.

#### 4.2.2 Trafic prioritaire à destination quelconque

Nous supposons maintenant que le trafic est généré par les feuilles est destiné à une feuille quelconque et que le séquençement des périodes d'activité alterne entre un séquençement favorisant un flux montant et un séquençement favorisant un flux descendant.

Nous prenons toujours le pire des cas, celui d'une trame reportée. En plus, nous considérons une trame qui doit être envoyée, après avoir subi le report et qu'elle soit reçue par le coordinateur de la feuille source, à contre courant avec le séquençement des périodes d'activité. Ce qui induit un délai supplémentaire d'un cycle global.

Ainsi, la formule de 4.2.1 devient :

$$2 * \text{duree\_cycle\_global} + [T_1; T_2] \leq 1 \text{seconde donc}$$

$$2 * ([T_0; T_1] + [T_1; T_2] + [T_2; T_3] + [T_3; T_0]) + [T_1; T_2] \leq 1$$

$[T_1; T_2] = n * (0,05 + 0,02) = n * 0,07$ , pour une topologie avec 8 feuilles actives maximum par coordinateur et 10 ms pour la durée de l'intervalle de relais garanti montant et 10 ms pour la durée de l'intervalle de relais garanti descendant.

En suivant le même raisonnement, pour une topologie de 8 feuilles actives maximum par coordinateur nous obtenons comme équation :  $0,00064 * n^2 + 0,366 * n - 1 = 0$ , qui a comme solution positive 2,7. Cela nous donne une topologie maximale avec 2 coordinateurs et 8 feuilles actives par coordinateur pour garantir un délai de bout-en-bout inférieur à 1 seconde pour toute trame envoyée d'une feuille à n'importe quelle autre feuille du réseau.

Pour le cas de 6 feuilles actives par coordinateur,  $[T_1; T_2] = n * (0,045 + 0,02) = n * 0,065$  et l'équation devient :  $0,00064 * n^2 + 0,341 * n - 1 = 0$  avec une solution positive égale à 2.9. Ainsi, nous obtenons une topologie maximale de 2 coordinateurs et 6 feuilles actives par coordinateur.

L'équation pour 4 feuilles actives par coordinateur est :  $0,00064 * n^2 + 0,266 * n - 1 = 0$  qui a comme solution positive 3,7. Ainsi, nous obtenons une topologie maximale de 3 coordinateurs avec 4 feuilles actives par coordinateur.

De la même façon, l'équation pour 2 feuilles actives par coordinateur est :  $0,00064 * n^2 + 0,216 * n - 1 = 0$  qui a comme solution positive 4,7. Ainsi, nous obtenons une topologie maximale de 4 coordinateurs avec 2 feuilles actives par coordinateur.

De la même manière, pour maximiser le délai de bout-en-bout à 2 secondes, la topologie maximale avec 8, 6, 4 et 2 feuilles actives par coordinateur nous donnent les 4 équations respectives :  $0,00064 * n^2 + 0,366 * n - 2 = 0$ ,  $0,00064 * n^2 + 0,341 * n - 2 = 0$ ,  $0,00064 * n^2 + 0,266 * n - 2 = 0$  et  $0,00064 * n^2 + 0,216 * n - 2 = 0$  respectivement. Cela nous donne 5,4, 5,8, 7,4 et 9 comme solutions respectives.

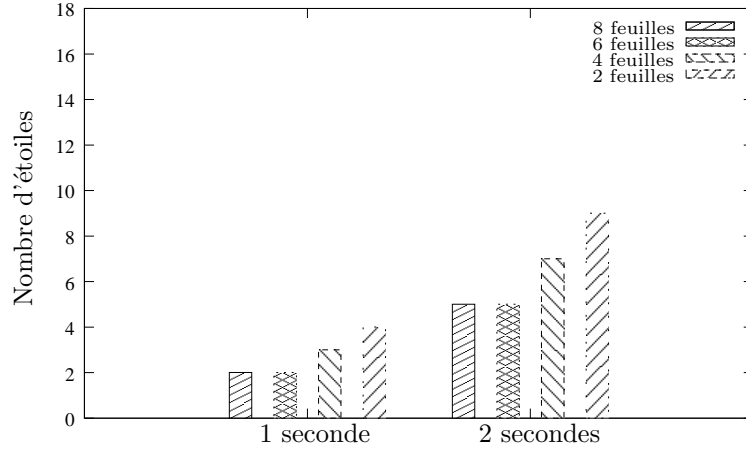


FIG. 4.42 – Taille du réseau avec  $[T_2; T_3] = [T_1; T_2]$  et un trafic à destination quelconque.

Notons que la taille du réseau s'est réduite à plus que la moitié quand la destination finale du trafic prioritaire est quelconque. Les hypothèses que nous avons retenues pour la taille des différents intervalles de temps nous conduisent à une configuration limitée à 2 étoiles pour garantir un délai de bout-en-bout qui ne dépasse pas la seconde.

#### 4.2.3 Influence de la durée de $[T_2; T_3]$

La durée de  $[T_2; T_3]$  affecte le délai de bout-en-bout. Dans les calculs précédents, nous avons pris comme durée de  $[T_2; T_3]$  la même que celle de  $[T_1; T_2]$ . Cette hypothèse pénalise le délai de bout-en-bout et donne un intervalle de temps largement suffisant pour que les coordinateurs échangent leur trafic de type non-prioritaire. Ainsi, pour favoriser le trafic de type prioritaire et lui garantir un délai plus petit, nous allons diviser la durée de  $[T_2; T_3]$  par 2, de plus, si ceci est fait au détriment du trafic non-prioritaire cela reste légitime.

En suivant le même raisonnement que celui décrit en 4.2.1 et 4.2.2, nous obtenons les équations récapitulées dans le tableau 4.3, pour un trafic à destination du CPAN pour 8, 6, 4 et 2 feuilles par coordinateur et un délai maximum de 1 et 2 secondes avec les solutions correspondantes :

En suivant la même démarche, les équations consignées dans le tableau 4.4 un trafic à destination quelconque pour 8, 6, 4 et 2 feuilles actives par coordinateur et un délai maximum de 1 et 2 secondes :

Les figures 4.43 et 4.44 nous permettent de remarquer qu'en divisant la durée de  $[T_2; T_3]$  par 2, nous pouvons atteindre des configurations plus grandes. Dans l'étude suivante nous allons considérer uniquement le CPAN comme puits de données (ce qui est le cas dans la plupart des applications) et observer l'effet de la durée de  $[T_2; T_3]$  sur le délai de bout-en-bout et la taille du réseau.

Nous considérons 4 durées différentes pour  $[T_2; T_3]$  :  $D_{T_1 T_2}/4$ ,  $D_{T_1 T_2}/2$ ,  $D_{T_1 T_2}$  et  $D_{T_1 T_2} * 2$ . Le choix des durées de  $[T_2; T_3]$  dépend des priorités relatives entre les deux types de trafic. Plus la durée de  $[T_2; T_3]$

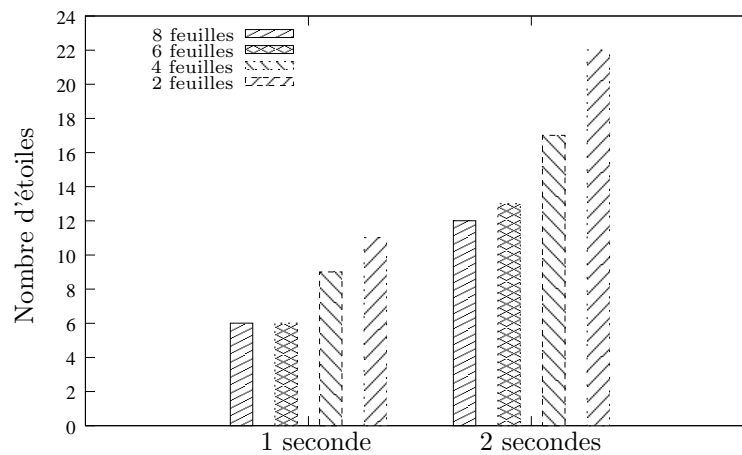


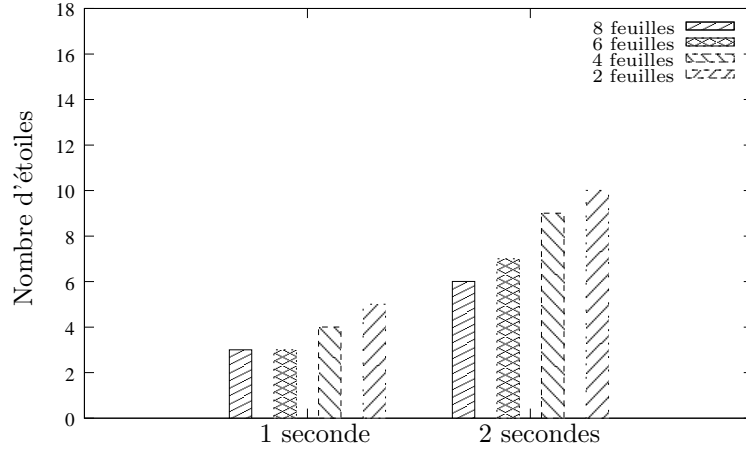
Nombre de feuilles	Délai max. (sec)	Équation	Solution
8	1	$0,00032 * n^2 + 0,158 * n - 1 = 0$	6
6	1	$0,00032 * n^2 + 0,145 * n - 1 = 0$	6
4	1	$0,00032 * n^2 + 0,108 * n - 1 = 0$	9
2	1	$0,00032 * n^2 + 0,083 * n - 1 = 0$	11
8	2	$0,00032 * n^2 + 0,158 * n - 2 = 0$	12
6	2	$0,00032 * n^2 + 0,145 * n - 2 = 0$	13
4	2	$0,00032 * n^2 + 0,108 * n - 2 = 0$	17
2	2	$0,00032 * n^2 + 0,083 * n - 2 = 0$	22

TAB. 4.3 – Équations pour le trafic à destination du CPAN.

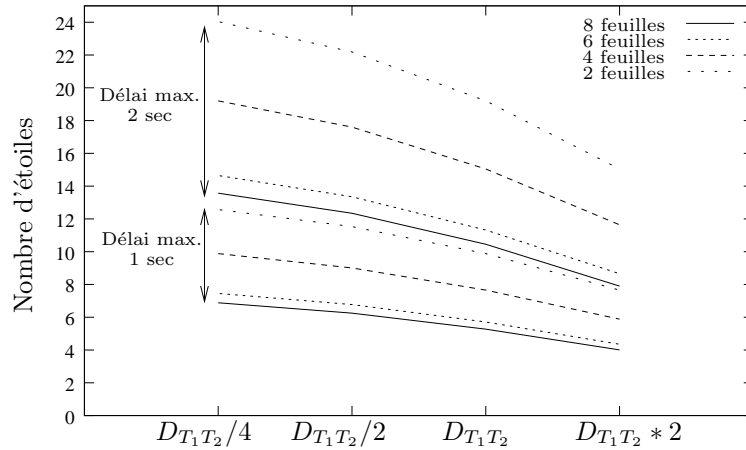
Nombre de feuilles	Délai max. (sec)	Equation	Solution
8	1	$0,00064 * n^2 + 0,296 * n - 1 = 0$	3
6	1	$0,00064 * n^2 + 0,276 * n - 1 = 0$	3
4	1	$0,00064 * n^2 + 0,216 * n - 1 = 0$	4
2	1	$0,00064 * n^2 + 0,176 * n - 1 = 0$	5
8	2	$0,00064 * n^2 + 0,296 * n - 2 = 0$	6
6	2	$0,00064 * n^2 + 0,276 * n - 2 = 0$	7
4	2	$0,00064 * n^2 + 0,216 * n - 2 = 0$	9
2	2	$0,00064 * n^2 + 0,176 * n - 2 = 0$	10

TAB. 4.4 – Équations pour le trafic à destination quelconque.

FIG. 4.43 – Taille du réseau avec  $[T_2; T_3] = [T_1; T_2]/2$  et un trafic à destination du CPAN.


 FIG. 4.44 – Taille du réseau avec  $[T_2; T_3] = [T_1; T_2]/2$  et un trafic à destination quelconque.

est petite, plus le trafic prioritaire est avantagé par rapport au trafic non prioritaire.


 FIG. 4.45 – Abaque : taille du réseau en fonction de la durée de  $[T_2; T_3]$ .

La figure 4.45 donne la taille maximale d'un réseau (en terme de nombre d'étoiles) qui garantit un délai de bout-en-bout maximum de 1 et 2 secondes pour un trafic à destination du CPAN. Ces résultats montrent qu'en pénalisant le trafic non-prioritaire (en diminuant la durée de  $[T_2; T_3]$  durant laquelle est routé le trafic non-prioritaire), nous pouvons garantir des meilleurs délais de bout-en-bout pour le trafic prioritaire.

#### 4.2.4 Mesures par simulation du délai de bout-en-bout pour deux *scenarii*

Dans cette partie, nous allons considérer deux *scenarii* qui correspondent aux deux courbes extrêmes de l'abaque du délai maximum d'une seconde : (i) un *scenario* avec 5 coordinateurs et 8 feuilles actives par coordinateur dont la topologie est donnée sur la figure 4.46 et (ii) un *scenario* avec 9 coordinateurs et 2 feuilles par coordinateur dont la topologie est donnée sur la figure 4.48.

Nous allons quantifier, par simulation, le délai de bout-en-bout pour les trames de type trafic priori-

taire générées par la feuille 44 et la feuille 26 pour chacun de ces deux *scenarii*. La durée des périodes d'activité est fixée selon les valeurs du tableau 4.2 (20 *ms* pour une étoile de 2 feuilles actives et 50 *ms* pour une étoile de 8 feuilles actives). Sur les figures 4.47 et 4.49 les droites verticales pointillées représentent le délai de bout-en-bout d'une seconde.

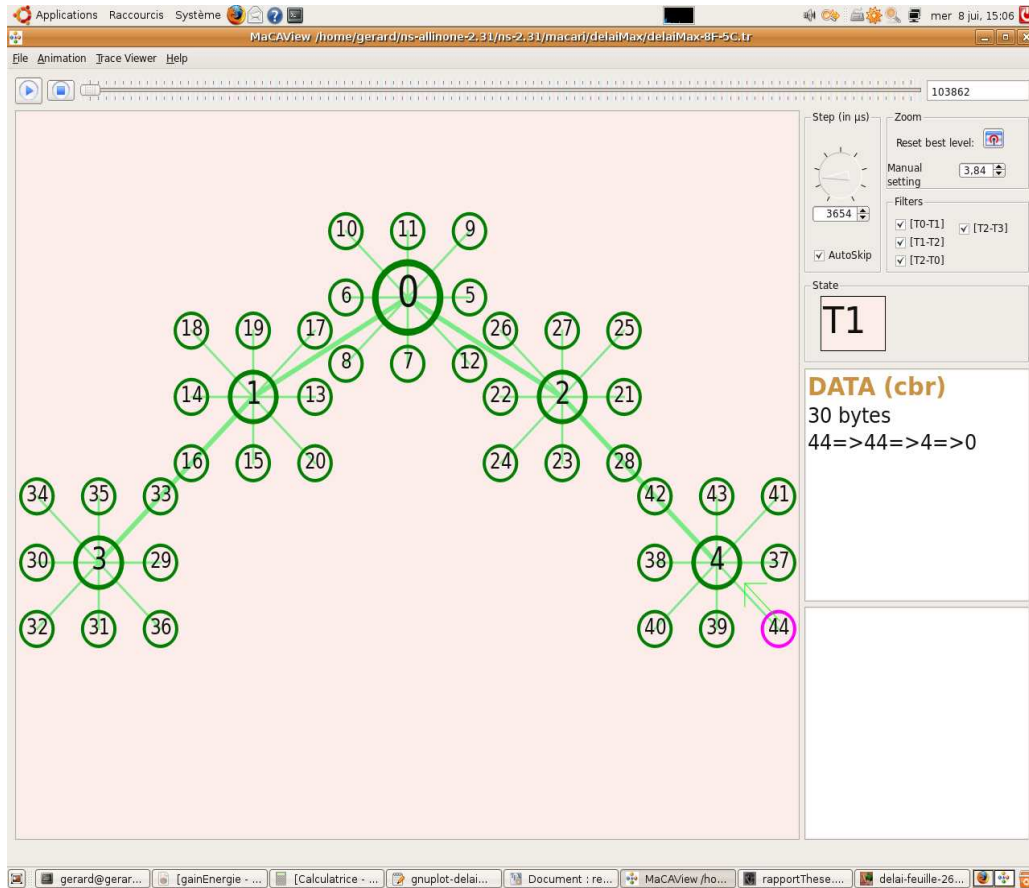


FIG. 4.46 – Une topologie de 5 coordinateurs avec 8 feuilles par coordinateur.

La figure 4.47 représente, sous forme de batons, toutes les trames de type prioritaire envoyées par la feuille 44 de la topologie 4.46 à destination du CPAN (le nœud 0 de la figure 4.46). Les résultats montrent que le délai de bout-en-bout du trafic de type prioritaire généré par cette feuille ne dépasse pas la seconde pour une topologie de 5 coordinateurs avec 8 feuilles actives par coordinateur. Par exemple, 20 trames ont été reçues avec un délai de bout-en-bout de 0,8 seconde.

Pour la feuille 26 de la topologie 4.48, la figure 4.49 montrent que toutes les trames de type prioritaire ont été envoyées à destination du CPAN sous un délai borné qui ne dépasse la seconde. Par exemple, 15 trames ont été reçues avec un délai de bout-en-bout de 0,5 seconde.

Ainsi, nous pouvons conclure que les résultats de simulation correspondent à nos calculs analytiques. Le délai de bout-en-bout ne dépasse pas la seconde pour les configurations que nous avons choisies parmi celles étudiées et représentées par l'abaque de la figure 4.45.

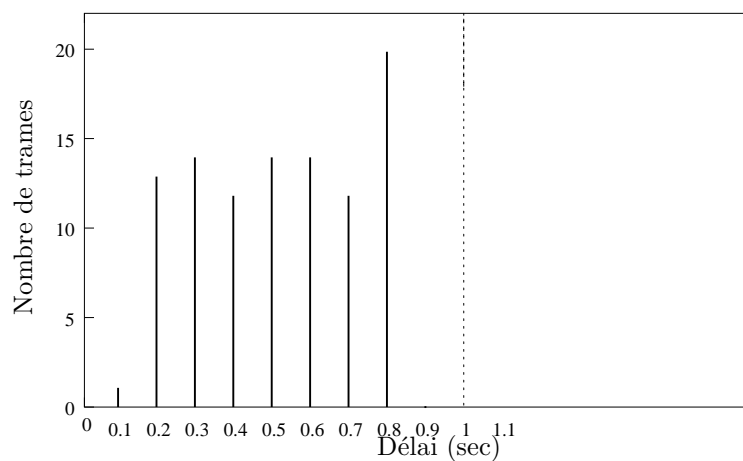


FIG. 4.47 – Distribution du délai de bout-en-bout pour la feuille 44 de la topologie 4.46.

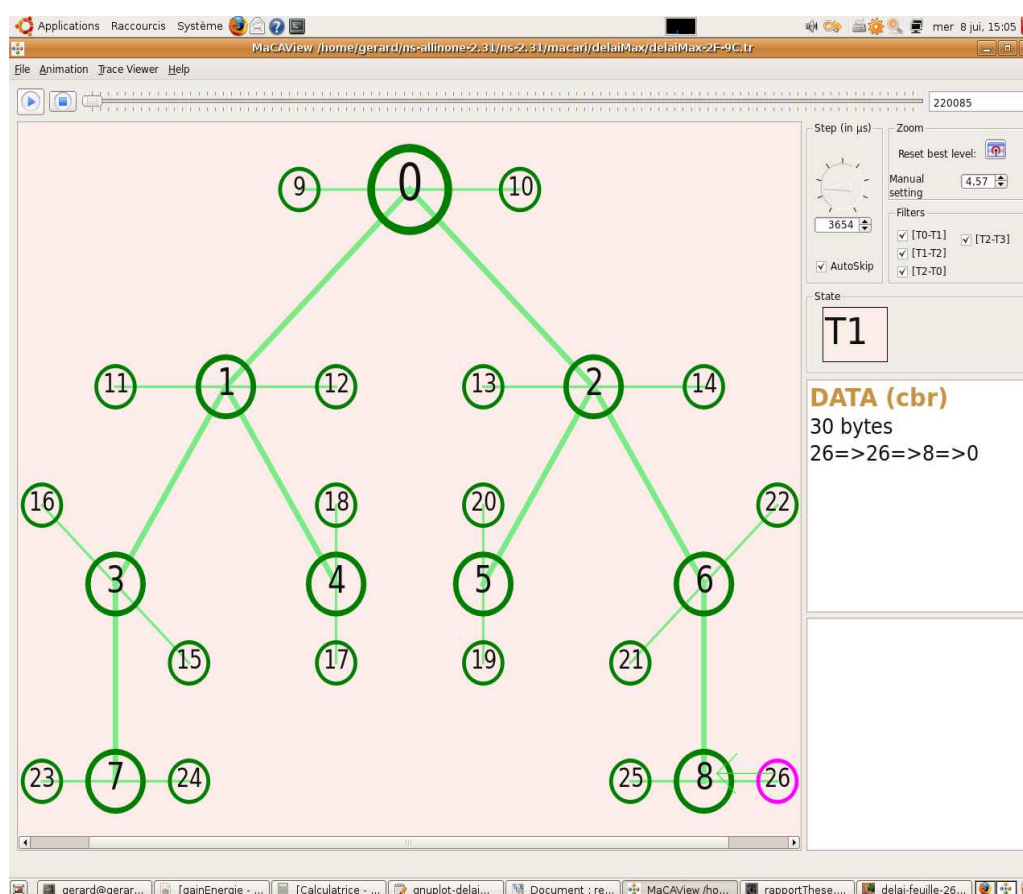


FIG. 4.48 – Une topologie de 9 coordinateurs avec 2 feuilles par coordinateur.

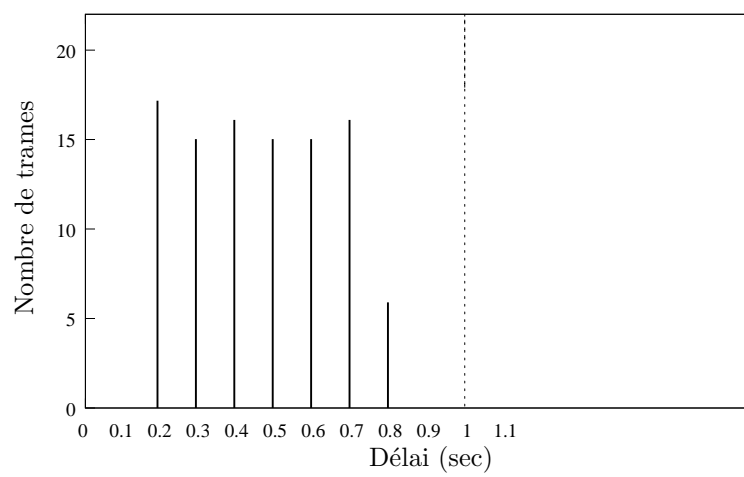


FIG. 4.49 – Distribution du délai de bout-en-bout pour la feuille 26 de la topologie 4.48.

## 5 Synthèse

Dans cette partie, nous concluons nos résultats en récapitulant comment MaCARI satisfait à nos objectifs en étant performant en terme d'économie d'énergie et de qualité de service. Concevoir un protocole MAC garantissant à la fois ces deux aspects revient à faire le juste compromis entre les deux, car il s'agit de deux aspects antagonistes.

Pour mettre en avant les caractéristiques de MaCARI qui offrent un fonctionnement économe en énergie, nous allons situer cette solution par rapport aux sources de consommation énergétique étudiées dans la partie 1.1.1 du chapitre 2. Les sources de consommation énergétique au niveau MAC sont essentiellement les collisions, l'*overhearing*, l'*idle listening*, la surcharge et les envois infructueux.

**Les collisions :** Grâce à la segmentation temporelle des activités des entités du réseau, MaCARI diminue le nombre de collisions dans le réseau. Ceci est constaté à partir des résultats présentés sur la figure 4.26. En effet, activer les entités du réseau par groupe diminue le nombre de concurrents pour accéder au canal et améliore les performances de CSMA/CA slotté, comme le montre les figures 4.26, 4.27 et 4.28. De plus, l'existence des intervalles de temps durant lesquels les échanges sont effectués en mode garanti diminue le nombre total de collisions dans le réseau.

**L'*idle listening* et l'*overhearing* :** En allouant à chaque étoile une période d'activité suffisante pour écouler le trafic de ses entités, MaCARI limite la durée de l'*idle listening* à la durée d'une période d'activité d'une étoile pour les feuilles. Il en est de même pour l'*overhearing* qui se limite pour les feuilles à l'écoute des échanges intra-étoile du trafic non-prioritaire.

**La surcharge :** MaCARI n'utilise pas d'algorithmes décentralisés qui nécessitent des échanges de proche en proche. La surcharge essentielle provient de la période de synchronisation. En revanche, durant cette période, la propagation du *beacon* est assurée sans risque de collision. Ainsi, la consommation énergétique des trames de contrôle est limitée à une seule trame physique par cycle pour maintenir la synchronisation des activités des différentes entités du réseau.

**Envois infructueux :** Le fait de définir un calendrier d'activités pour l'ensemble des entités du réseau permet d'éviter les envois infructueux. Les périodes d'activité de chaque entité sont connues et définies dans le *beacon*. Ainsi, MaCARI évite complètement la perte d'énergie due aux tentatives de communication avec des entités inactives.

Tout en garantissant un fonctionnement économe en énergie, MaCARI fournit deux types de qualité de service : le déterminisme et le délai borné de bout-en-bout.

**Le déterminisme :** Grâce aux GTS dans l'étoile et aux intervalles de relais garanti, MaCARI est capable d'assurer l'accès au médium pour un trafic de type prioritaire. Durant ces intervalles garantis, les échanges effectués ne sont pas gênés par d'autres communications dans le réseau et ne risquent pas de subir des interférences.

**Le délai de bout-en-bout :** MaCARI utilise la topologie arborescente du réseau pour établir un chemin de bout-en-bout garanti pour un trafic prioritaire. En adoptant alternativement un ordre favorisant les flux montants et un ordre favorisant les flux descendants, MaCARI borne le temps de traversée de bout-en-bout du réseau en 4 cycles globaux si le dimensionnement des intervalles est adapté à la charge du réseau.

## Chapitre 5

# Conclusion

Les réseaux de capteurs sans fil doivent satisfaire des besoins antagonistes comme l'économie d'énergie et la qualité de service. Trouver la meilleure solution pour garantir ces deux besoins à la fois ne semble pas être une option réaliste avec les technologies utilisées actuellement. Un compromis est donc à faire de façon à assurer un certain degré de qualité de service pour une catégorie de trafic tout en considérant l'aspect économie d'énergie.

Dans le cadre de cette thèse, nous avons étudié les méthodes et les protocoles les plus cités dans la littérature qui concernent l'accès au médium et qui sont dédiés aux réseaux de capteurs sans fils. La plupart de ces protocoles présentent des inconvénients importants : les hypothèses adoptées ne sont pas réalistes, les algorithmes sont trop complexes nécessitant de nombreux échanges et/ou de nombreux calculs, certains aspects sont limités. Notamment, pour réaliser un fonctionnement économe en énergie, la plupart des protocoles ont mis de côté la qualité de service.

Cela nous a amené à proposer MaCARI, un protocole MAC qui garantit à la fois l'économie d'énergie pour l'ensemble des entités du réseau et la qualité de service pour un certain type de trafic. Le mécanisme de MaCARI se base sur une étape de synchronisation durant laquelle un calendrier est diffusé à toutes les entités du réseau. Ce calendrier indique les périodes pendant lesquelles chaque entité est active et celles pendant lesquelles chaque entité est inactive. Ainsi, des rendez-vous entre les différents nœuds du réseau sont établis et le temps est découpé en cycles globaux périodiques annoncés par l'étape de synchronisation. Chaque cycle est découpé temporellement d'une manière à améliorer l'accès au médium et à garantir l'accès sans risque de collision durant des intervalles spécifiques, que nous avons appelés intervalles de relais. Le séquençement des intervalles à accès garanti permet de borner le temps de transit de l'information d'un bout à l'autre du réseau.

Le protocole MaCARI que nous proposons introduit un mécanisme emprunté à celui mis en place pour la migration des saumons. Certains poissons migrateurs, comme le saumon, ont besoin une fois



adulte, de rejoindre des frayères pour se reproduire. Pour cela ils doivent donc remonter une rivière à contre-courant en franchissant les obstacles rencontrés sur leur trajet quelque soit les courants rencontrés. Le franchissement de petites cascades ou de barrages édifiés par l'homme peut se révéler difficile voire impossible. Pour leur permettre de réussir à temps cette remontée indépendamment des flux rencontrés, un dispositif appelé échelle à saumon est souvent placé parallèlement à l'obstacle. Ce dispositif permet à un poisson de remonter « marche par marche » la différence de niveau, sans subir les conditions aléatoires d'une tentative de franchissement direct par un saut au dessus de l'obstacle à franchir. Les intervalles de relais garanti de MaCARI peuvent être vus comme les échelles à saumon. Grâce à ces intervalles de temps et leur séquençement, un trafic prioritaire (les saumons) est capable de remonter pas à pas la route vers la destination en suivant un chemin garanti éventuellement plus long et protégé sans risque de collision. Il est toutefois possible de tenter le chemin direct, mais sans garantie de succès.

Notre réflexion s'est accompagnée d'une démarche de simulation afin de vérifier le fonctionnement des différents mécanismes de MaCARI et de prouver leurs intérêts. Nous avons dû prendre en compte la modélisation du médium sans fil en implémentant un modèle de propagation qui se rapproche des conditions réelles de propagation. Suite à une série de mesures de portée des cartes de réseau sans fil sur lesquelles nous travaillons, nous avons implémenté le modèle ITU-R P1238-4, qui s'avère être le modèle le plus réaliste, en choisissant ses paramètres afin de s'adapter aux caractéristiques de l'environnement et des modules radio des cartes. Ce travail de simulation a nécessité l'implémentation sur NS2 des briques essentielles au fonctionnement de MaCARI comme la couche physique et le CSMA/CA slotté de la norme IEEE 802.15.4. Ainsi, et après avoir validé ces briques en les comparant à des résultats de mesures sur maquette, nous avons implémenté notre solution et étudié des améliorations, notamment pour la période de synchronisation, et ceci sur le simulateur mais aussi d'une façon analytique.

Afin de prouver la faisabilité de notre protocole, nous avons réalisé une maquette que nous avons implémentée sur des cartes de type B2400ZB-tiny. Les résultats issus des mesures sur cette maquette sont satisfaisants et convergent avec nos résultats de simulation. D'une part, cela valide nos modèles implémentés sur le simulateur NS2, et, d'autre part, prouve l'aspect réalisable de notre solution. La mise en place d'une maquette et la validation de MaCARI ont constitué une des étapes importantes du projet OCARI.

Ce travail a abouti à la mise en place et à la validation d'une méthode d'accès dédiée aux réseaux de capteurs sans fil. Cette méthode assure une qualité de service pour une catégorie de trafic et garantit un fonctionnement économe en énergie pour l'ensemble des entités du réseau. Les étapes franchies et les mécanismes étudiés durant cette thèse nous ont permis d'identifier plusieurs pistes de recherche à étudier. Dans la suite, nous citons les principales perspectives.

## Perspectives

Les techniques utilisées dans MaCARI ont inspiré la création d'autres thématiques dans l'équipe sur deux aspects : l'aspect cross-layering entre la couche MAC et la couche réseau en ce qui concerne le relais garanti, et l'aspect méthode d'accès hybride ayant des comportements qui diffèrent selon les intervalles de temps.

Plusieurs fonctionnalités ont été étudiées dans le cadre de cette thèse. D'autres pistes sont envisageables pour améliorer les performances de MaCARI, et cela à plusieurs niveaux et pour chacun des intervalles du cycle global, comme décrit dans la suite.

## Raccourcissement de la période de synchronisation

Nous avons présenté deux méthodes pour diminuer la durée de la période de synchronisation. D'autres méthodes peuvent être introduites pour améliorer d'avantage cette période. Le fait d'avoir un fils unique n'est pas souhaitable pour pouvoir appliquer des améliorations sur l'ordre de synchronisation. Un champ peut être renseigné dans le *beacon* pour indiquer le nombre actuel de fils coordinateurs. Ainsi, au moment de l'association, le nœud souhaitant s'associer au réseau peut choisir le coordinateur ayant déjà au moins un fils. De cette façon, l'arbre obtenu permet de créer un ordre de synchronisation plus convenable.

Durant la période de synchronisation, la diffusion des *beacons* est faite séquentiellement. Cela impose une durée de synchronisation qui augmente avec le nombre de coordinateurs dans le réseau. Grâce au coloriage des nœuds obtenu par SERENA, MaCARI peut permettre aux nœuds de la même couleur d'envoyer leurs *beacons* en même temps, en respectant toujours un ordre topologique.

Une autre piste consiste à ne diffuser le *beacon* de synchronisation qu'une fois tous les  $N$  cycles globaux. Ceci permettra de récupérer  $N - 1$  intervalles  $[T_0; T_1]$  pour d'autres activités.  $N$  dépend bien sûr des dérives observables.

## Découpage temporel de $[T_1; T_2]$ plus adapté

Ce que nous avons proposé pour le découpage temporel durant la période  $[T_1; T_2]$  est un séquençement temporel simple des périodes d'activité d'une même taille. Des améliorations peuvent être introduites à plusieurs niveaux pour améliorer l'utilisation du canal durant cette période. Il est intéressant d'appliquer un algorithme plus intelligent pour faire les allocations des périodes d'activité en fonction de la charge générée par chaque étoile par exemple.

Dans une topologie arborescente avec un trafic à destination de la racine, des goulots d'étranglement se créent sur les liens proches de la racine. Ce phénomène a des conséquences directes sur le délai de bout-en-bout pour le trafic de type prioritaire. Pour remédier à cela, le nombre de descendants d'un

coordinateur doit être pris en compte pour lui allouer une période d'activité adaptée. Nous pouvons supposer que le CPAN connaît les nœuds qui génèrent du trafic prioritaire et connaît aussi le volume de ce trafic. Un algorithme centralisé pourrait donc être utilisé.

Dans le cadre du projet OCARI, SERENA est proposé pour colorer les nœuds et les faire travailler durant le même slot de temps dans l'intervalle  $[T_2; T_3]$  quand ils ont la même couleur. SERENA (ou un autre protocole similaire) peut être appliqué pour colorer les étoiles et allouer aux étoiles de la même couleur la même période d'activité, ce qui permettrait de réduire  $[T_1; T_2]$ .

### **Optimisation de $[T_2; T_3]$**

Nous avons proposé pour méthode d'adressage l'adressage hiérarchique. Les adresses hiérarchiques sont assignées en fonction de la position d'un nœud dans l'arbre. Un changement de position, suite à une désassociation et une réassociation, conduit à un changement d'adresse d'un nœud. Cette instabilité d'adressage cause des conflits dans les tables de routage et nécessite des mécanismes de mises à jour dans les tables de correspondances entre adresses et entités au niveau applicatif. Conjointement avec notre partenaire INRIA, nous sommes en train de travailler sur une mécanique d'adressage qui offre une adresse stable de niveau 3 utilisée par le protocole de routage et une adresse hiérarchique de niveau 2.

Un conflit de couleur se produit quand deux nœuds de la même couleur s'empêchent ou empêchent un nœud tiers de recevoir correctement une trame à cause d'une collision. Des collisions de ce genre sont reproductibles si les conditions de la configuration restent stables. Avec notre partenaire INRIA, nous sommes en train de développer une méthode d'accès qui permet d'éviter ce genre de collisions. Ceci rend plus adaptatif le découpage TDMA du coloriage pour garantir un bon fonctionnement quand des conflits de coloriage se produisent, suite à une mobilité ou à une instabilité des liens. Cette méthode en cours d'étude est baptisée TDMA/CA.

# Nomenclature

AEA	Adaptative Election Algorithm	36
AI-LMAC	Adaptive, Information-centric and Lightweight MAC	37
ANR	Agence Nationale de Recherche	27
AODV	Ad hoc On-demand Distance Vector	66
APL	APplication sub-Layer	65
APS	APplication Support	66
B-MAC	Berkeley-MAC	43
BE	Backoff Exponent	55
BI	Beacon Interval	53
BLE	Battery Life Extension	57
BO	Beacon Order	53
CAP	Contention Access Period	53
CCA	Clear Channel Assessment	39
CFP	Contention Free Period	53
CPAN	Coordinateur du PAN	72
CRC	Cyclic Redundancy Check	75
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance	34
CSMA/CD	Carrier Sense Multiple Access with Collision Detection	38
CTS	Clear To Send	41
CW	Contention Window	55
D-MAC	Data gathering MAC	43

DCF Distributed Coordination Function .....	39
DCN Direction de Constructions Navales .....	27
DIFS DCF InterFrame Spacing .....	40
DRAND Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks .....	44
DW-LPL Dual Wake-up LPL .....	44
E-MAC Event-MAC .....	37
EDF Électricité De France .....	27
EMPR Energy efficient MPR .....	78
EOLSR Energy-aware OLSR .....	77
FFD Full Function Device .....	49
FLAMA FLow-Aware Medium Access .....	36
FRTS Future Request To Send .....	42
G-MAC Gateway-MAC .....	46
GPS Global Positionning System .....	24
GTIM Gateway Traffic Indication Message .....	46
GTS Guaranteed Time Slot .....	53
IEEE Institute of Electrical and Electronics Engineers .....	57
IHM Interface Homme-Machine .....	114
INRIA Institut National de Recherche en Informatique et Automatique .....	27
ISF Inter Frame Space .....	54
ITU International Telecommunication Union .....	105
L-MAC Lightweight-MAC .....	37
LATTIS Laboratoire Toulousain de Technologie et d'Ingénierie des Systèmes .....	27
LIMOS Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes .....	27
LPL Low Power Listening .....	43
LQI Link Quality Indication .....	52
LR-WPAN Low Rate Wireless Personal Area Networks .....	49

LRI Laboratoire de Recherche en Informatique .....	27
MAC Medium Access Control .....	55
MaCARI Mac pour oCARI .....	71
MLDE MAC Layer Data Entity .....	50
MLME MAC Layer Management Entity .....	50
MPR MultiPoint Relay .....	78
NAM Natwork AniMator .....	104
NB Number of Backoffs .....	55
NIB Network layer Information Base .....	63
NLDE Network Layer Data Entity-Service Access Point .....	50
NLME Network Layer Management Entity .....	50
NP Neighbor Protocol .....	35
NS2 Network Simulator 2 .....	104
OCARI Optimisation des Communications Ad-hoc pour les Réseaux Industriels .....	27
OLSR Optimized Link State Routing .....	77
OSI Open Systems Interconnection .....	104
PAN Personal Area Network .....	49
PCF Point Coordination Function .....	39
PD PHY layer Data .....	51
PHY Physical layer .....	55
PLME Physical Layer Management Entity .....	51
RFD Reduced Function Device .....	49
RTS Request To Send .....	41
S-MAC Sensor-MAC .....	41
SAP Service Access Point .....	50
SD Superframe Duration .....	53
SEP Schedule Exchange Protocol .....	35

SERENA SchEdule RoutEr Nodes Activity .....	78
SIFS Short Inter Frame Spacing .....	40
SO Superframe Order .....	53
T-MAC Timeout-MAC .....	42
TDMA Time Division Multiple Access .....	34
TRAMA TRaffic-Adaptive Medium Access control .....	35
WLAN Wireless Local Area Network .....	104
WPAN Wireless Personnal Area Network .....	104
Z-MAC Zebre-MAC .....	44
ZDO ZigBee Device Object .....	65

# Appendices





## Annexe A

# Identification de la zone d'interférences

La zone d'interférences est à l'origine de perturbations qui affectent le fonctionnement de CSMA/CA ainsi que les protocoles de coloriage niveau 3. Cette zone est comprise entre le seuil de réception et le seuil de détection de porteuse. Le seuil de détection de porteuse est toujours à  $-95\text{ dBm}$  indépendamment du débit de transmission, alors que le seuil de réception est lui fonction du débit. Pour la couche physique IEEE 802.11 avec un débit de  $11\text{ Mbits/s}$  le seuil de réception est généralement (ceci dépend légèrement de la qualité du matériel utilisé) de  $-82\text{ dBm}$ , alors que pour la couche physique IEEE 802.15.4 avec un débit de  $250\text{ Kbits/s}$  le seuil de réception est de  $-92\text{ dBm}$ . Le but de cette série de mesures est d'identifier la distance qui sépare les deux seuils de la norme IEEE 802.15.4.

Pour réaliser cela, nous allons essayer de mettre en avant l'effet de cette zone sur le fonctionnement de CSMA/CA. Nous avons eu besoin de 3 cartes pour effectuer nos mesures :

- une carte qui transmet une série 1000 trames à la suite en CSMA/CA avec une période d'émission de  $5\text{ ms}$ . Nous désignons cette carte par la carte CSMA/CA,
- une carte réceptrice jouant le rôle d'un superviseur. Nous désignons cette carte par le superviseur,
- une carte qui émet en continu avec une période d'émission de  $4\text{ ms}$  des trames sans appliquer le CSMA/CA. Nous désignons cette carte par la carte perturbatrice.

Les 3 cartes sont placées comme le montre la figure A.1. La carte CSMA/CA est placée à  $30\text{ cm}$  du superviseur. Les trois cartes sont positionnées à  $1\text{ m}$  du sol. Nous allons faire déplacer la carte perturbatrice d'une position initiale dans laquelle les deux autres cartes (la carte CSMA/CA et le superviseur) sont à sa portée jusqu'à une position où les deux autres cartes sont hors de sa portée. La trajectoire de la carte perturbatrice suit la médiatrice du segment correspondant à la position des deux cartes fixes.

Durant ce déplacement, 3 cas de figures peuvent se présenter :

1. le superviseur reçoit les trames des deux cartes avec des retards ou des pertes de trames provenant des deux cartes émettrices. Dans ce cas, nous pouvons déduire que la carte perturbatrice est à portée de la carte CSMA/CA et du superviseur,
  2. le superviseur reçoit uniquement les trames CSMA/CA avec des retards. Dans ce cas, nous pouvons déduire que la carte perturbatrice gêne la carte CSMA/CA en occupant le canal et lui causant des tirages de *backoff* sans que les trames provenant de la carte perturbatrice soient reçues par le superviseur. Dans ce cas, nous pouvons constater que les carte CSMA/CA et superviseur se situent dans la zone d'interférence de la carte perturbatrice,
  3. le superviseur reçoit uniquement les trames provenant de la carte CSMA/CA sans retard ni perte. Dans ce cas, nous pouvons déduire que la carte perturbatrice est hors portée des deux autres cartes.
- Ces trois cas 1, 2 et 3 sont représentés sur la figure A.1.

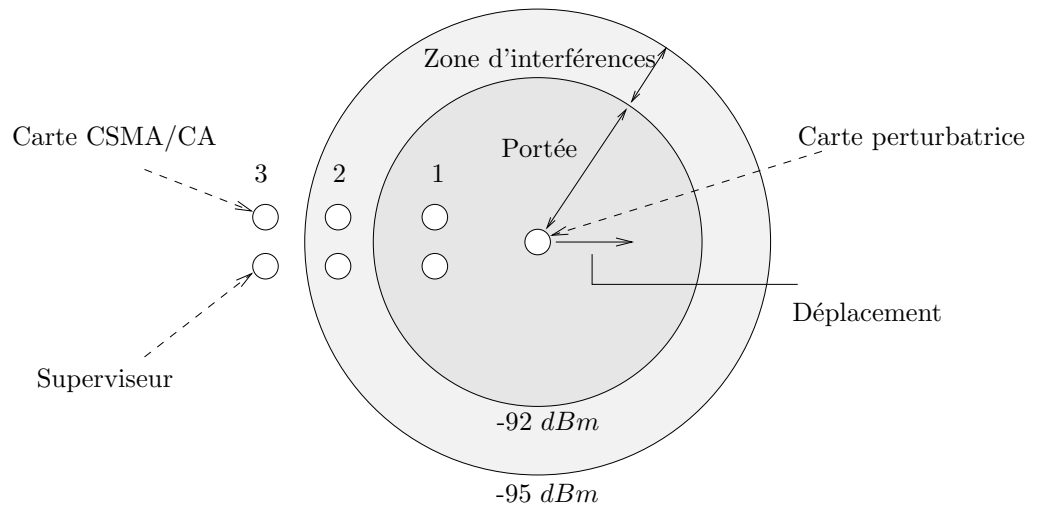


FIG. A.1 – Position des cartes CSMA/CA et superviseur par rapport à la carte perturbatrice.

Nous avons observé le nombre de trames reçues envoyées par la carte perturbatrice. Nous pouvons remarquer sur le figure A.2 que le nombre de trames perdues augmente avec la distance. Cependant, entre 68 m et 78 m ce nombre subit de variations qui sont certainement dues aux conditions de propagation, en limite de portée, de l'endroit dans lequel nous avons effectué nos mesures.

Nous avons constaté qu'en même temps que le nombre de trames reçues de la carte perturbatrice est nul (à une distance entre 78 et 82 m), nous avons eu des pertes de trames et des retards d'accès au médium pour les trames envoyées en CSMA/CA. Ceci est représenté sur la figure A.3 sur laquelle nous avons affiché ce qui se passe dans les derniers mètres de la portée de la carte perturbatrice.

Les résultats que nous avons obtenus ne nous permettent pas de conclure que ce retard était dû à la présence de la carte CSMA/CA dans la zone d'interférence de la carte perturbatrice, parce que

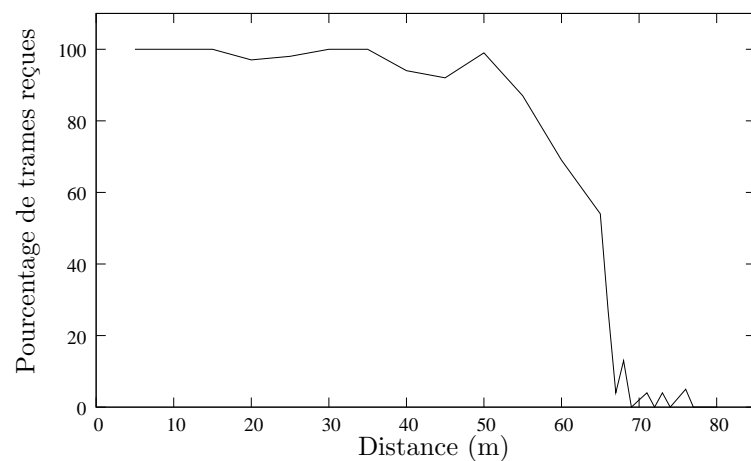


FIG. A.2 – Nombre de trames reçues en fonction de la distance.

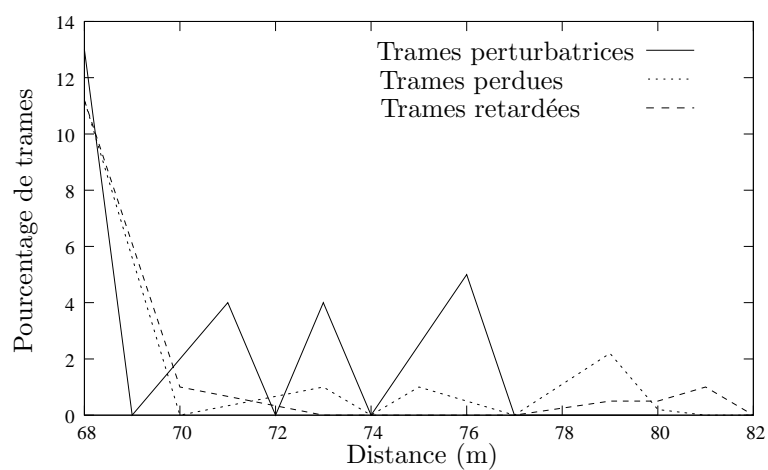


FIG. A.3 – Nombre de trames perdues ou retardées envoyées par la carte CSMA/CA.

la portée était tellement variable que nous ne pouvons pas être certains que la carte superviseur et la carte CSMA/CA se trouvaient toutes les deux dans cette zone. En d'autres termes, nous ne pouvons pas considérer que si le superviseur ne reçoit pas de trame de la carte perturbatrice, la carte CSMA/CA ne les recevrait pas non plus. Il se peut que le superviseur soit hors portée de la carte perturbatrice et ne reçoit donc pas ses trames, alors que la carte CSMA/CA soit à portée de la carte perturbatrice.

D'autre part, ce qui nous semble évident, suite à cette expérimentation, est le fait que cette zone dans les réseaux de capteurs sans fil avec un débit de  $250\text{ Kbits/s}$  ne doit pas être considérée comme étant deux fois la portée (estimation usuelle pour 802.11) car dans notre cas elle est beaucoup plus petite.

## Annexe B

# Format des trames MaCARI

### Format général d'une trame MaCARI

Le format général d'une trame MaCARI est donné sur le tableau B.1.

<b>Octets : 2</b>	<b>1</b>	<b>0/2</b>	<b>0/2</b>	<b>0/2/8</b>	<b>0/2</b>	<b>0/2/8</b>	<b>variable (114 max)</b>	<b>2</b>
<i>Frame control</i>	<i>Sequence number</i>	<i>Dest. PAN ID</i>	<i>Dest. ad- dress</i>	<i>Final dest. address</i>	<i>Src. PAN ID</i>	<i>Src. ad- dress</i>	<i>Payload</i>	FCS
MaCARI header							MaCARI payload	MaCARI footer

TAB. B.1 – Format général d'une trame MaCARI.

Par la suite nous détaillons chaque champ de la trame.

### Le champ frame control

Le champ frame control est détaillé dans le tableau B.2.

<b>Bits : 0-2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10-11</b>	<b>12-13</b>	<b>14-15</b>
<i>Frame type</i>	<i>Security enabled</i>	<i>Frame pen- ding</i>	<i>Ack- Request/ Coloring- Mode</i>	<i>PAN ID com- pres- sion</i>	<i>trafic- Type</i>	<i>Priority</i>	<i>Color indica- tion</i>	<i>Dest. adres- sing mode</i>	<i>Frame compa- tibility</i>	<i>Src. adres- sing mode</i>

TAB. B.2 – Le champ frame control

**Frame type :** le tableau B.3 spécifie les valeurs de frame type selon le type de la trame.

Valeur de frame type	Description
000	trame de beacon
001	trame de données
010	trame d'acquittement
011	trame de command MAC
100-111	réservés

TAB. B.3 – Les types de trames

**Ack request/Coloring Mode :** pour une trame de données, ce champ est utilisé pour indiquer si la trame envoyée doit être acquittée ou pas en mettant le bit à 1. Dans le cas d'une de beacon, en mettant ce bit à 1, ce champ indique au coordinateurs qu'il faut lancer l'algorithme de coloriage.

**PAN ID compression bit :** si le bit du PAN ID compression est égale à 1 et les deux adresses source et destination sont présentes, seulement le Dest. PAN ID sera présent et le Src PAN ID sera considéré identique à Dest. PAN ID. Si juste une des deux adresses est présente ce bit est mis à 0 et le PAN ID de l'adresse présente est spécifié. Si aucune adresse n'est présente ce bit est mis à 0. Le tableau B.4 récapitule les valeurs admises en fonction des adresses présentes.

PAN ID compression bit	Dest. address	Src. address	Dest. PAN ID	Src. PAN ID
1	présente	présente	présent	non présent
1	présente	non présente	non valide	non valide
1	non présente	présente	non valide	non valide
1	non présente	non présente	non valide	non valide
0	présente	présente	présent	présent
0	présente	non présente	présent	non présent
0	non présente	présente	non présent	présent
0	non présente	non présente	non présent	non présent

TAB. B.4 – Le bit PAN ID compression

**trafficType :** deux types de trafic ont été considérés dans OCARI, un trafic prioritaire à relayer en suivant l'arbre durant  $[T_1; T_2]$ , et un trafic non-prioritaire à router durant  $[T_2; T_3]$ . Si ce bit est mis à 1, la trame doit être routée sur l'arbre dans les intervalles de relais garantis. Si le bit est mis à 0, la trame sera routée durant la période  $[T_2; T_3]$ .

**Priority :** deux niveaux de priorité supplémentaires ont été considérés dans OCARI pour chacun des deux types de trafic. Ceci permet une meilleure gestion des files d'attente de chaque type de trafic. Si ce bit est mis à 1, la trame est considérée prioritaire et 0 dans le cas contraire.

**Color indication :** ce bit est utilisé pour une trame de beacon uniquement pour indiquer la prise en compte du coloriage de SERENA. Si ce bit est mis à 1, le payload du beacon contient les deux champs

Color sequence et Colored slot duration, une valeur de 0 indique l'absence de ces deux champs.

**Dest. addressing mode et Src. addressing mode :** ces deux champs spécifient de type d'adressage utilisé pour les adresses destination et source respectivement. Les valeurs de ces deux champs sont listées dans le tableau B.5.

Valeur de l'addressing mode	Description
00	le PAN ID et les champs d'adressage ne sont pas présents
01	Réservé
10	Le champ d'adresse contient une adresse courte de 2 octets
11	Le champ d'adresse contient une adresse longue de 8 octets

TAB. B.5 – Addressing modes

**Frame compatibility :** ce champ indique de type de protocole encapsulé dans la trame MAC.

### Le champ Sequence number

Ce champ spécifie le numéro de séquence de la trame au niveau MAC. Le numéro de séquence des beacons est indépendant de celui des trames de données ou de commande.

### Le champ Dest. PAN ID

Ce champ spécifie l'identifiant du PAN auquel est adressée la trame. La valeur 0xffff correspond à l'identifiant de diffusion et est accepté par toutes les entités écoutant le médium.

### Le champ Dest. address

Ce champ spécifie l'adresse destination du prochain saut de la trame sur 2 ou 8 octets selon le mode spécifié dans le tableau B.5. La valeur 0xffff correspond à l'adresse de diffusion.

### Le champ Final dest. address

Ce champ spécifie l'adresse destination finale de la trame. Il permet à MaCARI d'effectuer le relais sur l'arbre.

### Le champ Src. PAN ID

Ce champ spécifie l'identifiant du PAN duquel provient la trame.



### **Le champ Src. address**

Ce champ spécifie l'adresse source de la trame sur 2 ou 8 octets selon le mode spécifié dans le tableau B.5.

### **Le champ Payload**

Ce champ est de taille variable et son contenu varie selon le type de la trame.

### **Le champ FCS**

Le mécanisme de détection d'erreurs est le même que celui de la norme IEEE 802.15.4 [38].

## Annexe C

# Liste des primitives de MaCARI

Ce travail a été réalisé avec la contribution de l'INRIA et du LATTIS.

## 1 Primitives de données

Le tableau C.1 liste les primitives de données MaCARI.

Nom	Request	Indication	Confirm
MaCARI-DATA	1.1	1.2	1.3

TAB. C.1 – Liste des primitives de données

### 1.1 MaCARI-DATA.request

**Description :** Des messages de données des couches supérieures peuvent parvenir à MaCARI par l'intermédiaire de la couche NwCARI. Ces paquets de données peuvent être des messages applicatifs ou des paquets de contrôle propres à EOLSR ou à SERENA. Tous les paquets de données qui parviennent à MaCARI par EOLSR possèdent un type qui est "contraint" ou "non contraint".

**Sémantique :** MaCARI-DATA.request (SrcAddrMode, DstAddrMode, DstPANId, DstAddr, FinalDstAddr, msduLength, msdu, msduHandle, TxOption).

Les champs de cette primitive sont spécifiés sur le tableau C.2.

### 1.2 MaCARI-DATA.indication

**Description :** MaCARI remonte à la couche NwCARI les messages de données de type non-contraint par l'arbre, et tout message diffusé ou dont la destination finale est la station elle-même.

Nom	Type	Valeurs	Description
SrcAddrMode		0x00-0x03	indique le type d'adresse à employer pour l'adresse source, soit l'adresse courte (0x02), soit l'adresse longue (0x03), soit sans adresse (0x00)
DstAddrMode		0x00-0x03	indique le type d'adresse à employer pour l'adresse destination, soit l'adresse courte (0x02), soit l'adresse longue (0x03), soit sans adresse (0x00)
DstPANId		2 octets	indique l'indentifiant du PAN auquel appartient l'entité destinatrice
DstAddr		2/8 octets	indique l'adresse destination selon le format indiqué dans DstAddrMode
FinalDstAddr		2 octets	indique l'adresse courte de la destination finale, ce champ ne fait pas partie de l'entête réseau
msduLength		1 octet	indique la taille des données à envoyer
msdu	suite d'octets		la suite des octets à envoyer dans la trame
msduHandle	entier		sert pour identifier la suite d'octets à envoyer (qui est le msdu)
TxOption	bitmap	1 octet	TX_OPT_COLORING_MODE 0x80 (uniquement pour le beacon), TX_OPT_COLORATION 0x40 (uniquement pour le beacon), TX_OPT_PRIORITY 0x20, TX_OPT_TRAFFIC_TYPE 0x10 (1 pour contrainte), TX_OPT_SECURITY_ENABLE 0x08, TX_OPT_INDIRECT 0x04, TX_OPT_GTS 0x02, TX_OPT_ACK_REQ 0x01, TX_OPT_NONE 0x00
Timestamp	entier	5 octets	à la création du paquet, ce champ est renseigné avec une valeur nulle par la couche réseau. Ainsi, la couche MAC estampille la trame au niveau deux. Si ce champ est renseigné par la couche réseau, la couche MAC estampille la trame avec la valeur de ce champ.

TAB. C.2 – Champs de MaCARI-DATA.request

**Sémantique :** MaCARI-DATA.indication (SrcAddrMode, SrcPANId, SrcAddr, DstAddrMode, DstPANId, DstAddr, TrafficType, Priority, msduLength, msdu, LinkQuality/RSSI, Timestamp, ReceptionTimestamp).

Les champs de cette primitive sont spécifiés sur le tableau C.3.

### 1.3 MaCARI-DATA.confirm

**Description :** Il s'agit d'une confirmation d'envoi de données suite à un appel de MaCARI-DATA.request.

**Sémantique :** MaCARI-DATA.confirm (Status, msduHandle, Timestamp)

Les champs de cette primitive sont spécifiés sur le tableau C.4.

## 2 Primitives de contrôle

Le tableau C.5 liste les primitives de contrôle MaCARI.

### 2.1 MaCARI-ASSOCIATION.request

**Description :** appelée par la couche Management pour demander à MaCARI de lancer la procédure d'association. Le choix du coordinateur auquel on souhaite s'associer n'est pas spécifié pour l'instant. Par défaut, on demande l'association au coordinateur dont le beacon a été reçu en premier.

**Sémantique :** MaCARI-ASSOCIATION.request (LogicalChannel, CoordAddrMode, CoordPANId, CoordAddr, CapabilityInformation)

Les champs de cette primitive sont spécifiés sur le tableau C.6.

### 2.2 MaCARI-ASSOCIATION.indication

**Description :** appelée par MaCARI pour indiquer à la couche Management que MaCARI vient d'accepter une association d'un nouveau fils en indiquant son adresse courte.

**Sémantique :** MaCARI-ASSOCIATION.indication (ShortAddr, CapabilityInformation)

Les champs de cette primitive sont spécifiés sur le tableau C.7

### 2.3 MaCARI-ASSOCIATION.confirm

**Description :** envoyée par MaCARI à la couche Management pour lui donner le résultat de la requête d'association.

**Sémantique :** MaCARI-ASSOCIATION.confirm (AssocShortAddr, ParentDepth, Status)

Les champs de cette primitive sont spécifiés sur le tableau C.8.

Nom	Type	Valeurs	Description
SrcAddrMode	entier	0x00-0x03	indique le type d'adresse à employer pour l'adresse source, soit l'adresse courte (0x02), soit l'adresse longue (0x03), soit sans adresse (0x00)
SrcPANId	entier	2 octets	indique l'identifiant du PAN auquel appartient la source du message
SrcAddr		2/8 octets	indique l'adresse source du message selon le format indiqué dans SrcAddrMode
DstAddrMode	entier	0x00-0x03	indique le type d'adresse à employer pour l'adresse destination, soit l'adresse courte (0x02), soit l'adresse longue (0x03), soit sans adresse (0x00)
DstPANId	entier		indique l'identifiant du PAN auquel appartient l'entité destinatrice
DstAddr		2/8 octets	indique l'adresse destination selon le format indiqué dans DstAddrMode
TrafficType	bool	1 bit	précise le type de trafic envoyé, trafic contraint (0x01) ou trafic non-contraint (0x00)
Priority	bool	1 bit	précise la priorité du message, soit prioritaire (0x01) soit non-prioritaire (0x00)
msduLength	entier	1 octet	indique la taille des données reçues
msdu	suite d'octets		la suite des octets reçus dans la trame
LinkQuality/RSSI	entier	2 octets	indique la qualité du lien et la puissance reçue du signal
Timestamp	entier	5 octets	le temps auquel le message a été créé au noeud originaire de la trame. 1 octet pour désigner le numéro de séquence du cycle global et 4 octets pour indiquer le nombre de périodes de backoff consommées depuis T1 du cycle indiqué.
Reception-Timestamp	entier	5 octets	le temps de réception de la trame au niveau MAC. Ce champ est présent dans le cas où la station est la destination finale de la trame pour le trafic temps contraint, et pour toutes les remontées du trafic temps non-contraint. 1 octet pour désigner le numéro de séquence du cycle global et 4 octets pour indiquer le nombre de périodes de backoff consommées depuis T1 du cycle indiqué.

TAB. C.3 – Champs de MaCARI-DATA.indication

Nom	Type	Valeurs	Description
Status	Enumération	SUCCESS, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, FRAME_TOO_LONG (cf. la norme)	le statut du dernier msdu envoyé
msduHandle	entier		le handle associé au msdu qui doit être envoyé
Timestamp	entier		le temps auquel le msdu a été envoyé, il est rempli et pris en compte juste dans le cas SUCCESS

TAB. C.4 – Champs de MaCARI-DATA.confirm

## 2.4 MaCARI-ASSOCIATION.response

**Description :** envoyée par MaCARI d'un coordinateur ou PAN coordinateur à l'entité qui essaie de s'associer.

**Sémantique :** MaCARI-ASSOCIATION.response (DeviceAddr, AssocShortAddr, ParentDepth, Status)

Les champs de cette primitive sont spécifiés sur le tableau C.9.

## 2.5 MaCARI-ASSOCIATION-NOTIFICATION.request

**Description :** appelée par MaCARI pour notifier MaCARI du coordinateur PAN d'une nouvelle association. Cette primitive est à l'initiative du nouveau coordinateur entrant.

**Sémantique :** MaCARI-ASSOCIATION-NOTIFICATION.request (NewCoordAddress, ParentCoordAddress, NewCoordDepth)

Les champs de cette primitive sont spécifiés sur le tableau C.10.

## 2.6 MaCARI-ASSOCIATION-NOTIFICATION.indication

**Description :** envoyée par MaCARI d'un coordinateur pour informer MaCARI du coordinateur PAN de son association.

**Sémantique :** MaCARI-ASSOCIATION-NOTIFICATION.indication (NewCoordAddress, ParentCoordAddress, NewCoordDepth)

Les champs de cette primitive sont spécifiés sur le tableau C.11

Nom	Req- uest	Ind- ication	Resp- onse	Conf- irm	NwCARI	Manag- ement	Interne
MaCARI-ASSOCIATION	2.1	2.2	2.4	2.3	-	X	-
MaCARI-ASSOCIATION-NOTIFICATION	2.5	2.6	-	-	-	-	X
MaCARI-DIRECT-ASSOCIATION	2.7	-	-	2.8	-	X	-
MaCARI-GET	2.9	-	-	2.10	-	X	-
MaCARI-SET	2.11	-	-	2.12	-	X	-
MaCARI-DISASSOCIATION	2.13	2.14	-	2.15	-	X	-
MaCARI-DISASSOCIATION-NOTIFICATION	2.16	2.17	-	-	-	-	X
MaCARI-GTS	2.20	2.22	-	2.21	-	X	-
MaCARI-SCAN	2.18	-	-	2.19	-	X	-
MaCARI-NEIGHBOR-COLORS	2.23	-	-	2.24	X	-	-
MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST	2.25	-	-	2.26	X	-	-
MaCARI-LINK-CHARACTERISTICS-LIST	2.27	-	-	2.28	-	X	-
MaCARI-CONFLICT	-	2.29	-	-	X	-	-
MaCARI-SYNC-LOSS	-	2.30	-	-	-	X	-
MaCARI-LINK-LOSS	-	2.31	-	-	X	-	-
MaCARI-TRX-STATE	-	2.32	-	-	-	X	-
MaCARI-COLORING-MODE-ON	2.33	2.35	-	2.34	X	-	-
MaCARI-MAX-COLOR	2.36	-	-	2.37	X	-	-

TAB. C.5 – Liste des primitives de contrôle

Nom	Type	Valeurs	Description
LogicalChannel			le numéro du canal que la couche APS a choisi
CoordAddrMode			le mode d'adressage pour l'adresse du coordinateur avec qui on veut s'associer
CoordPANId			l'identifiant du PAN auquel appartient le coordinateur avec qui on veut s'associer
CoordAddr			l'adresse du coordinateur avec qui on veut s'associer
Capability-Information			précise les informations sur les capacités de l'entité. Capacité à devenir PAN coordinateur ou pas, capacité à être coordinateur ou routeur ou end-Device, alimentation...

TAB. C.6 – Champs de MaCARI-ASSOCIATION.request

Nom	Type	Valeurs	Description
ShortAddr			l'adresse courte du fils
Capability-Information			précise les informations sur les capacités de l'entité. Capacité à devenir PAN coordinateur ou pas, capacité à être coordinateur ou routeur ou end-Device, alimentation...

TAB. C.7 – Champs de MaCARI-ASSOCIATION.indication

Nom	Type	Valeurs	Description
AssocShortAddr	entier		indique l'adresse courte allouée par le coordinateur avec qui l'entité s'est associée. Vaut 0xFFFF si l'association a échoué
ParentDepth	entier	1 octet	indique la profondeur dans l'arbre du coordinateur père. Vaut 0xFF si l'association a échoué
Status	Enumération	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG (cf. la norme)	le statut du dernier msdu envoyé

TAB. C.8 – Champs de MaCARI-ASSOCIATION.confirm



Nom	Type	Valeurs	Description
DeviceAddr	64-bit IEEE address	toute adresse de type 64-bit IEEE address	l'adresse longue de l'entité qui a demandé l'association
AssocShortAddr	entier		indique l'adresse courte allouée par le coordinateur. Vaut 0xFFFF si l'association a échoué
ParentDepth	entier	1 octet	indique la profondeur dans l'arbre du coordinateur père. Vaut 0xFF si l'association a échoué
Status	Enumération	0x00-0x02	0x00 Association Successful, 0x01 PAN complet, 0x02 refus d'accès au PAN

TAB. C.9 – Champs de MaCARI-ASSOCIATION.response

Nom	Type	Valeurs	Description
NewCoordAddress	short address	2 octets	l'adresse courte du coordinateur entrant
ParentCoord-Address	short address	2 octets	l'adresse courte du coordinateur père
NewCoordDepth	entier	1 octet	la profondeur dans l'arbre du coordinateur entrant

TAB. C.10 – Champs de MaCARI-ASSOCIATION-NOTIFICATION.request

Nom	Type	Valeurs	Description
NewCoordAddress	short address	2 octets	l'adresse courte du coordinateur entrant
ParentCoord-Address	short address	2 octets	l'adresse courte du coordinateur père
NewCoordDepth	entier	1 octet	la profondeur dans l'arbre du coordinateur entrant

TAB. C.11 – Champs de MaCARI-ASSOCIATION-NOTIFICATION.indication

## 2.7 MaCARI-DIRECT-ASSOCIATION.request

**Description :** appelée par la couche Management pour demander à MaCARI d'accepter une association d'une manière forcée (sans réception de requête d'association).

**Sémantique :** MaCARI-DIRECT-ASSOCIATION.request (LongAddress, CapabilityInformation)

Les champs de cette primitive sont spécifiés sur le tableau C.12

Nom	Type	Valeurs	Description
LongAddress	64-bit IEEE address	toute adresse de type 64-bit IEEE address	l'adresse longue de l'entité
Capability-Infomation	bitmap		précise les informations sur les capacités de l'entité. Capacité à devenir PAN coordinateur ou pas, capacité à être coordinateur ou routeur ou end-Device, alimentation...

TAB. C.12 – Champs de MaCARI-DIRECT-ASSOCIATION.request

## 2.8 MaCARI-DIRECT-ASSOCIATION.confirm

**Description :** générée par MaCARI pour répondre à une MaCARI-DIRECT-ASSOCIATION.request soumise par la couche Management.

**Sémantique :** MaCARI-DIRECT-ASSOCIATION.confirm (Status, DeviceAddress)

Les champs de cette primitive sont spécifiés sur le tableau C.13

Nom	Type	Valeurs	Description
Status		SUCCESS, ALREADY_PRESENT, MAX_CHILDREN_REACHED (cf. la norme)	indique si l'entité s'est bien associée ou pas
DeviceAddress	64-bit IEEE address	toute adresse de type 64-bit IEEE address	l'adresse longue de l'entité qui figurait dans la MaCARI-DIRECT-ASSOCIATION.request

TAB. C.13 – Champs de MaCARI-DIRECT-ASSOCIATION.confirm

## 2.9 MaCARI-GET.request

**Description :** appelée par la couche Management pour obtenir la valeur d'un attribut de la couche MaCARI. La table C.18 contient la liste des attributs de MaCARI avec les valeurs correspondantes.

**Sémantique :** MaCARI-GET.request (AttributID)

Les champs de cette primitive sont spécifiés sur le tableau C.14

Nom	Type	Valeurs	Description
AttributID		voir table C.18	l'identifiant de l'attribut à modifier

TAB. C.14 – Champs de MaCARI-GET.request

## 2.10 MaCARI-GET.confirm

**Description** : générée par MaCARI pour confirmer la lecture d'un attribut suite à la soumission de MaCARI-GET.request.

**Sémantique** : MaCARI-GET.confirm (Status, AttributID)

Les champs de cette primitive sont spécifiés sur le tableau C.15

Nom	Type	Valeurs	Description
Status		SUCCESS, UNSUPPORTED_ATTRIBUTE ou INVALID_INDEX	renvoie le résultat de la MaCARI-SET.request avec SUCCESS si l'attribut est bien mis à jour ou bien la raison pour laquelle cela n'est pas fait
AttributID		voir table C.18	l'identifiant de l'attribut à modifier

TAB. C.15 – Champs de MaCARI-GET.confirm

## 2.11 MaCARI-SET.request

**Description** : appelée par la couche Management pour mettre à jour un attribut de la couche MaCARI. La table C.18 contient la liste des attributs de MaCARI avec les valeurs correspondantes.

**Sémantique** : MaCARI-SET.request (AttributID, Value)

Les champs de cette primitive sont spécifiés sur le tableau C.16

Nom	Type	Valeurs	Description
AttributID		voir table C.18	l'identifiant de l'attribut à modifier
Value		voir table C.18	la valeur à fixer

TAB. C.16 – Champs de MaCARI-SET.request

## 2.12 MaCARI-SET.confirm

**Description** : générée par MaCARI pour confirmer la mise à jour ou pas d'un attribut suite à un appel de MaCARI-SET.request.

**Sémantique** : MaCARI-SET.confirm (Status, AttributID)

Les champs de cette primitive sont spécifiés sur le tableau C.17

Nom	Type	Valeurs	Description
Status		SUCCESS, READ_ONLY, UNSUPPORTED_ATTRIBUTE, INVALID_INDEX, INVALID_VALUE	renvoie le résultat de la MaCARI-SET.request avec SUCCESS si l'attribut est bien mis à jour ou bien la raison pour laquelle cela n'est pas fait
AttributID		voir table C.18	l'identifiant de l'attribut à modifier

TAB. C.17 – Champs de MaCARI-SET.confirm

Attribut	Id	Type	Valeur	Description
Adresse longue MAC	0x00	adresse IEEE	adresse longue IEEE sur 64 bits	
Adresse courte	0x01	Entier	0x0000-0xffff	
Adresse longue MAC du père	0x02	adresse IEEE	adresse longue IEEE sur 64 bits	
Adresse courte du père	0x03	Entier	0x0000-0xffff	
Type de l'entité	0x04		COORD, ROUTER ou END-DEVICE	
Durée d'activité	0x05	Entier		en terme de période de backoff, modifiable par l'applicaton au niveau du CPAN uniquement
Durée du cycle global	0x06	Entier		en terme de période de backoff, modifiable par l'applicaton au niveau du CPAN uniquement
Rm	0x07	Entier		le nombre de routeurs/coordonateurs maximum par coordinateur
Cm	0x08	Entier		le nombre de fils (routeur/coordonateur/end-device) maximum par coordinateur
Lm	0x09	Entier		indique la profondeur maximum de l'arbre
Durée d'un slot coloré	0x0A	Entier		en terme de période de backoff, modifiable par l'applicaton au niveau du CPAN uniquement

TAB. C.18 – Attributs de MaCARI (à compléter)

### 2.13 MaCARI-DISASSOCIATION.request

**Description :** appelée par la couche Management pour demander la désassociation de l'entité elle-même, ou bien de l'un de ses fils.

**Sémantique :** MaCARI-DISASSOCIATION.request (DeviceAddr, DisassociateReason)

Les champs de cette primitive sont spécifiés sur le tableau C.19

Nom	Type	Valeurs	Description
DeviceAddr		adresse IEEE	l'adresse longue de l'entité à informer
Disassociation-Reason			la raison pour laquelle on veut cette désassociation, soit l'entité veut se désassocier soit un PAN coordinateur veut que l'entité se désassocie

TAB. C.19 – Champs de MaCARI-DISASSOCIATION.request

### 2.14 MaCARI-DISASSOCIATION.indication

**Description :** appelée par MaCARI pour indiquer à la couche Management que MaCARI vient de désassocier un fils en indiquant l'adresse courte de ce fils.

**Sémantique :** MaCARI-DISASSOCIATION.indication (ShortAddr, DisassociateReason)

Les champs de cette primitive sont spécifiés sur le tableau C.20

Nom	Type	Valeurs	Description
DeviceAddr		adresse IEEE	adresse longue de l'entité souhaitant se désassocier
DisassociateReason			la raison pour la désassociation

TAB. C.20 – Champs de MaCARI-DISASSOCIATION.indication

### 2.15 MaCARI-DISASSOCIATION.confirm

**Description :** initiée par MaCARI pour confirmer le résultat de de MaCARI-DISASSOCIATION.request générée par la couche Management.

**Sémantique :** MaCARI-DISASSOCIATION.confirm (Status, DeviceAddr)

Les champs de cette primitive sont spécifiés sur le tableau C.21

### 2.16 MaCARI-DISASSOCIATION-NOTIFICATION.request

**Description :** appelée par MaCARI pour notifier MaCARI du coordinateur du PAN de la désassociation d'un coordinateur fils. Cette primitive est à l'initiative du père du coordinateur qui sort du réseau.

Nom	Type	Valeurs	Description
Status		SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG (cf. la norme)	SUCCESS si la MaCARI-DISASSOCIATION.request est acquittée par le coordinateur père, ou une des autres valeurs citées si ce n'est pas le cas
DeviceAddr			adresse longue de l'entité souhaitant se désassocier

TAB. C.21 – Champs de MaCARI-DISASSOCIATION.confirm

**Sémantique :** MaCARI-DISASSOCIATION-NOTIFICATION.request (LeavingCoordAddress, ParentCoordAddress)

Les champs de cette primitive sont spécifiés sur le tableau C.22

Nom	Type	Valeurs	Description
LeavingCoord-Address	short address	2 octets	l'adresse du coordinateur sortant
ParentCoord-Address	short address	2 octets	l'adresse du père du coordinateur sortant

TAB. C.22 – Champs de MaCARI-DISASSOCIATION-NOTIFICATION.request

## 2.17 MaCARI-DISASSOCIATION-NOTIFICATION.indication

**Description :** appelée par MaCARI pour indiquer à MaCARI du coordinateur du PAN que MaCARI vient d'accepter un départ d'un coordinateur de l'arbre.

**Sémantique :** MaCARI-DISASSOCIATION-NOTIFICATION.indication (LeavingCoordAddress, ParentCoordAddress)

Les champs de cette primitive sont spécifiés sur le tableau C.23

Nom	Type	Valeurs	Description
LeavingCoord-Address	short address	2 octets	l'adresse courte du coordinateur sortant
ParentCoord-Address	short address	2 octets	l'adresse courte du coordinateur père

TAB. C.23 – Champs de MaCARI-DISASSOCIATION-NOTIFICATION.indication

## 2.18 MaCARI-SCAN.request

**Description :** appelée par la couche Management pour effectuer un SCAN.

**Sémantique :** MaCARI-SCAN.request (ScanType, ScanChannels, ScanDuration, ChannelPage)

Les champs de cette primitive sont spécifiés sur le tableau C.24

Nom	Type	Valeurs	Description
ScanType			le mode de scan à employer : passif, actif, orphan, ED
ScanChannels	bitmap		bitmap sur 27 bits (27 canaux) indiquant la liste des canaux à scanner ; 1= SCAN 0= no SCAN
ScanDuration	entier	0-14	la durée du scan sur chaque canal, $48 * (2^n + 1)$ symboles, avec $n$ la valeur de ScanDuration
ChannelPage	entier		le channel page sur lequel il faut faire le scan

TAB. C.24 – Champs de MaCARI-SCAN.request

## 2.19 MaCARI-SCAN.confirm

**Description :** envoyée par MaCARI à la couche Management pour l'informer du résultat du scan suite à un appel de MaCARI-SCAN.request.

**Sémantique :** MaCARI-SCAN.confirm (status, ScanType, ChannelPage, UnscannedChannels, ResultListSize, EnergyDetectList, PANDescriptorList)

Les champs de cette primitive sont spécifiés sur le tableau C.25

Nom	Type	Valeurs	Description
status			le résultat du scan
ScanType			le mode de scan à employer : 0x00 ED, 0x01 scan actif, 0x02 scan passif, 0x03 orphan scan
ChannelPage	entier		à compléter
Unscanned-Channels	bitmap		bitmap sur 27 bits (27 canaux) indiquant la liste des canaux qui n'ont pas été scanés ; 1= SCAN 0= no SCAN, cet attribut n'est pas valide pour un scan d'énergie
ResultListSize	entier		le nombre d'éléments dans le liste de résultat, il est mis à 0 pour l'orphan scan
EnergyDetectList	liste d'entiers		le niveau d'énergie mesuré pour chaque canal, uniquement pour un scan d'énergie
PANDescriptorList	liste de valeurs de PAN descriptors	à venir liste des valeurs d'un PAN descriptor (se référer à la norme)	la liste de PAN descriptors pour chaque beacon reçu, cet attribut est nul pour un scan d'énergie ou un orphan scan

TAB. C.25 – Champs de MaCARI-SCAN.confirm

## 2.20 MaCARI-GTS.request

**Description :** appelée par les couches supérieures la couche Management pour demander à MaCARI de faire une requête de GTS.

**Sémantique :** MaCARI-GTS.request (GTSCaracteristics)

Les champs de cette primitive sont spécifiés sur le tableau C.26.

Nom	Type	Valeurs	Description
GTS-Characteristics	GTS characteristics		les caractéristiques du GTS

TAB. C.26 – Champs de MaCARI-GTS.request

## 2.21 MaCARI-GTS.confirm

**Description :** envoyée par MaCARI à la couche Management pour lui donner le résultat de la requête de GTS.

**Sémantique :** MaCARI-GTS.confirm (GTSCaracteristics, Status)

Les champs de cette primitive sont spécifiés sur le tableau C.27.

Nom	Type	Valeurs	Description
GTS-Characteristics	GTS characteristics		les caractéristiques du GTS
Status	Enumération	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG (cf. la norme)	le statut du dernier msdu envoyé

TAB. C.27 – Champs de MaCARI-GTS.confirm

## 2.22 MaCARI-GTS.indication

**Description :** appelée par MaCARI pour indiquer à la couche Management que MaCARI vient de recevoir une demande de GTS.

**Sémantique :** MaCARI-GTS.indication (DeviceAddress, GTSCaracteristics)

Les champs de cette primitive sont spécifiés sur le tableau C.28



Nom	Type	Valeurs	Description
DeviceAddress	short address	2 octets	l'adresse de la feuille qui a fait la requête
GTS-Characteristics	GTS characteristics		les caractéristiques du GTS

TAB. C.28 – Champs de MaCARI-GTS.indication

## 2.23 MaCARI-NEIGHBOR-COLORS.request

**Description** : appelée par la couche NwCARI pour indiquer à MaCARI les couleurs des voisins pour rester éveillé durant les slots consacrés à ces couleurs.

**Sémantique** : MaCARI-NEIGHBOR-COLORS.request (NodeColor, NumberOfColors, ColorList)

Les champs de cette primitive sont spécifiés sur le tableau C.29

Nom	Type	Valeurs	Description
NodeColor			la couleur du noeud lui-même
NumberOfColors			le nombre de couleurs dans mon voisinage
ColorList			liste des couleurs de mes voisins

TAB. C.29 – Champs de MaCARI-NEIGHBOR-COLORS.request

## 2.24 MaCARI-NEIGHBOR-COLORS.confirm

**Description** : envoyée par MaCARI pour confirmer la prise en compte de MaCARI-NEIGHBOR-COLORS.request.

**Sémantique** : MaCARI-NEIGHBOR-COLORS.confirm (status)

Les champs de cette primitive sont spécifiés sur le tableau C.30

Nom	Type	Valeurs	Description
status			à compléter

TAB. C.30 – Champs de MaCARI-NEIGHBOR-COLORS.confirm

## 2.25 MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.request

**Description** : appelée par NwCARI pour récupérer la liste des entités entendues.

**Sémantique** : MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.request ()

## 2.26 MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.confirm

**Description** : envoyée par MaCARI à la couche NwCARI en réponse à MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.request.

**Sémantique :** MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.confirm (numberOfNeighbors, unidirectionalNeighborList)

Les champs de cette primitive sont spécifiés sur le tableau C.31

Nom	Type	Valeurs	Description
numberOfNeighbors	entier		le nombre de voisins entendus par MaCARI
unidirectionalNeighborList			la liste des adresses courtes des entités entendues par MaCARI

TAB. C.31 – Champs de MaCARI-UNIDIRECTIONAL-NEIGHBOR-LIST.confirm

## 2.27 MaCARI-LINK-CHARACTERISTICS-LIST.request

**Description :** appelée par la couche Management pour récupérer la liste des RSSI et LQI des liens radio des voisins coordinateurs et de ses feuilles. Pour une feuille, seul le lien avec son coordinateur sera obtenu en retour.

**Sémantique :** MaCARI-LINK-CHARACTERISTICS-LIST.request ()

## 2.28 MaCARI-LINK-CHARACTERISTICS-LIST.confirm

**Description :** envoyée par MaCARI à la couche Management en réponse à MaCARI-LINK-CHARACTERISTICS-LIST.request.

**Sémantique :** MaCARI-LINK-CHARACTERISTICS-LIST.confirm (NumberOfNeighbors, AddressNeighborList, RSSINeighborList, LQINeighborList)

Les champs de cette primitive sont spécifiés sur le tableau C.32

Nom	Type	Valeurs	Description
NumberOfNeighbors	entier		le nombre de voisins entendus par MaCARI
AddressNeighborList			la liste des adresses courtes des entités entendues par MaCARI
RSSINeighborList			la liste des RSSI respectifs
LQINeighborList			la liste des LQI respectifs

TAB. C.32 – Champs de MaCARI-LINK-CHARACTERISTICS-LIST.confirm

## 2.29 MaCARI-CONFLICT.indication

**Description :** envoyée par MaCARI à NwCARI pour l'informer d'un conflit de couleur.

**Sémantique :** MaCARI-CONFLICT.indication (SrcShortAddr, DstShortAddr, ConflictType)

Les champs de cette primitive sont spécifiés sur le tableau C.33

Nom	Type	Valeurs	Description
SrcShortAddr		2 octets	l'adresse source du message en conflit
DstShortAddr		2 octets	l'adresse destination du message en conflit
ConflictType			à compléter

TAB. C.33 – Champs de MaCARI-CONFLICT.indication

### 2.30 MaCARI-SYNC-LOSS.indication

**Description :** envoyée par MaCARI à la couche NwCARI pour l'informer que l'entité a perdu la synchronisation avec le père, ou pour l'informer d'une désassociation de l'un de ses fils.

**Sémantique :** MaCARI-SYNC-LOSS.indication (Type, Address, Reason)

Les champs de cette primitive sont spécifiés sur le tableau C.34

Nom	Type	Valeurs	Description
Type			trois types de désynchronisation : une désassociation initiée par l'entité elle-même, une désassociation forcée par le père de l'entité ou bien une rupture de lien
PANID			l'identifiant du PAN duquel l'entité s'est désynchronisée
LogicalChannel			le canal sur lequel était synchronisée l'entité

TAB. C.34 – Champs de MaCARI-SYNC-LOSS.indication

### 2.31 MaCARI-LINK-LOSS.indication

**Description :** envoyée par MaCARI à la couche NwCARI pour l'informer qu'un lien avec un noeud voisin donné vient de disparaître.

**Sémantique :** MaCARI-LINK-LOSS.indication (ShortAddr)

Les champs de cette primitive sont spécifiés sur le tableau C.35

Nom	Type	Valeurs	Description
ShortAddr		2 octets	l'adresse courte du voisin avec qui le lien est perdu

TAB. C.35 – Champs de MaCARI-LINK-LOSS.indication

### 2.32 MaCARI-TRX-STATE.indication

**Description :** envoyée par MaCARI à la couche Management pour l'informer de l'état du module radio.

**Sémantique :** MaCARI-TRX-STATE.indication (State, Durée)

Les champs de cette primitive sont spécifiés sur le tableau C.36

Nom	Type	Valeurs	Description
State	Enumération	RX_ON, TX_ON, TRX_OFF	indique le nouvel état du module radio
Durée	entier		la durée en terme de periode de backoff de l'état précédent

TAB. C.36 – Champs de MaCARI-TRX-STATE.indication

### 2.33 MaCARI-COLORING-MODE-ON.request

**Description :** cette primitive est appelée par la couche NwCARI du coordinateur du PAN pour informer MaCARI qu'il faut mettre le bit Coloring Mode à 1 dans le prochain beacon à envoyer.

**Sémantique :** MaCARI-COLORING-MODE-ON.request ()

### 2.34 MaCARI-COLORING-MODE-ON.confirm

**Description :** cette primitive est envoyée par MaCARI du coordinateur du PAN à la couche NwCARI pour lui confirmer la prise en compte de MaCARI-COLORING-MODE-ON.request.

**Sémantique :** MaCARI-COLORING-MODE-ON.confirm (Status)

Les champs de cette primitive sont spécifiés sur le tableau C.37

Nom	Type	Valeurs	Description
Status		SUCCESS ou ERROR	indique le résultat de la prise en compte de la demande de mise à jour du champ Coloring Mode

TAB. C.37 – Champs de MaCARI-COLORING-MODE-ON.confirm

### 2.35 MaCARI-COLORING-MODE-ON.indication

**Description :** cette primitive est appelée par MaCARI, suite à la réception d'un beacon avec le champ Coloring Mode à 1, pour informer la couche NwCARI qu'il faut lancer l'algorithme de coloriage.

**Sémantique :** MaCARI-COLORING-MODE-ON.indication ()

### 2.36 MaCARI-MAX-COLOR.request

**Description :** cette primitive est appelée par la couche NwCARI du coordinateur du PAN pour informer MaCARI que le coloriage dans le réseau est terminé en lui passant le nombre maximum de couleur en

paramètre. Par conséquent, à la réception de cette primitive le champ ColoringMode du prochain beacon sera mis à 0, le champ ColorIndication sera mis à 1, le champ ColorSequence sera affecté la valeur de MaxColor.

**Sémantique :** MaCARI-MAX-COLOR.request (MaxColor)

Les champs de cette primitive sont spécifiés sur le tableau C.38

Nom	Type	Valeurs	Description
MaxColor	Integer		indique le nombre maximum de couleur dans le réseau

TAB. C.38 – Champs de MaCARI-MAX-COLOR.request

## 2.37 MaCARI-MAX-COLOR.confirm

**Description :** cette primitive est envoyée par MaCARI du coordinateur du PAN à la couche NwCARI pour lui confirmer la prise en compte de MaCARI-MAX-COLOR.request.

**Sémantique :** MaCARI-MAX-COLOR.confirm (Status)

Les champs de cette primitive sont spécifiés sur le tableau C.39

Nom	Type	Valeurs	Description
Status		SUCCESS ou ERROR	indique le résultat de la prise en compte de la demande de mise à jour du champ Coloring Mode

TAB. C.39 – Champs de MaCARI-MAX-COLOR.confirm

# Liste des publications internationnales

1. Gérard Chalhoub, Alexandre Guitton, François Delobel, Michel Misson, "Implementation details for the lower layers of an LP-WPAN simulation framework", dans *IJWCS (International Journal of Wireless Communication and Simulation)*, à paraître.
2. Khaldoun Al Agha, Gérard Chalhoub, Alexandre Guitton, Erwan Livolant, Saoucène Mahfoudh, Pascale Minet, Michel Misson, Joseph Rahmé, Thierry Val, Adrien Van Den Bossche, "Cross-layering in an industrial wireless sensor network : case study of OCARI", dans *JNW (Journal of Networks)*, Décembre 2009.
3. Gérard Chalhoub, Nassima Haddid, Alexandre Guitton, Michel Misson, "Deference mechanisms significantly increase the MAC delay of slotted CSMA/CA", dans *ICC (International Conference on Communications)*, Juin 2009.
4. Gérard Chalhoub, Erwan Livolant, Alexandre Guitton, Adrien van den Bossche, Michel Misson, Thierry Val, "Specifications and evaluation of a MAC protocol for a LP-WPAN", dans *AHSWN (Ad Hoc and Sensor Wireless Networks – An International Journal)*, Décembre 2008.
5. Marc-Henry Bertin, Adrien van den Bossche, Gérard Chalhoub, Tuan Dang, Saoucène Mahfoudh, Joseph Rahmé, Jean-Baptiste Viollet, "OCARI for industrial wireless sensor networks", dans *IFIP Wireless Days*, Novembre 2008.
6. Gérard Chalhoub, Alexandre Guitton, Frédérique Jacquet, Antonio Freitas, Michel Misson, "Medium Access Control for a Tree-Based Wireless Sensor Network : Synchronization Management", dans *IFIP Wireless Days*, Novembre 2008.
7. Gérard Chalhoub, Alexandre Guitton, Michel Misson, "MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part A : MaCARI : a synchronized tree-based MAC protocol", dans *WSAN (IFIP Conference on Wireless Sensor and Actor Networks)*, Juillet 2008.

8. T. Dang, C. Devic, and al., "OCARI : Optimization of communication for ad hoc reliable industrial networks," dans *IEEE INDIN*, Janvier 2008.
9. Gérard Chalhoub, Antonio Freitas, Michel Misson, "A Novel Approach for Simulating a Bio-Contamination Process", dans *BIODEVICES (International Conference on Biomedical Electronics and Devices)*, Janvier 2008.

# Bibliographie

- [1] K. Romer and F. Mattern, “The design space of wireless sensor networks,” in *IEEE Wireless Communications*, 2006.
- [2] “The 29 palms experiment : Tracking vehicles with a uav-delivered sensor network,” <http://robotics.eecs.berkeley.edu/pister/29Palms0103/>.
- [3] G. Simon, A. Ledezczi, and M. Maroti, “Sensor network-based countersniper system,” in *Sensys*, november 2004.
- [4] J. Hart and K. Martinez, “Environmental sensor networks : A revolution in the earth system science?” in *Elsevier B.V*, 2006.
- [5] H. Baldus, K. Klabunde, and G. Muesch, “Reliable set-up of medical body-sensor networks,” in *EWSN 2004*, January 2004.
- [6] C. Kappler and G. Riegel, “A real-world, simple wireless sensor network for monitoring electrical energy consumption,” in *EWSN 2004*, January 2004.
- [7] F. Michahelles, P. Matter, A. Schmidt, and B. Schiele, “Applying wearable sensors to avalanche rescue,” in *Computers and graphics*, 2003.
- [8] C. Baouche, A. Freitas, and M. Misson, “Radio proximity detection in a WSN to localize mobile entities within a confined area,” *Journal of Communications*, vol. 4, no. 4, May 2009.
- [9] T. Dang, C. Devic, and al., “OCARI : Optimization of communication for ad hoc reliable industrial networks,” in *IEEE INDIN*, January 2008.
- [10] The OCARI project, “The OCARI project web site,” <http://ocari.lri.fr>.
- [11] Agence Nationale de Recherche, “The ANR web site,” <http://www.agence-nationale-recherche.fr/>.
- [12] A. Jayasuriya, S. Perreau, A. Dadej, and S. Gordon, “Hidden vs. exposed terminal problem in ad hoc networks,” in *Proceedings of the Australian Telecommunication Networks and Applications Conference*, December 2004.
- [13] Telit, “Telit,” <http://www.telit.com/>.



- [14] V. Rajendran, K. Obraczka, and J. Gracia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” in *ACM Sensys*, Los Angeles, California, USA, November 2003.
- [15] V. Rajendran, J. Garcia-Luna-Aceves, and K. Obraczka, “Energy-efficient, application-aware medium access for sensor networks,” in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, Washington, USA, November 2005.
- [16] T. Nieberg, S. Dulman, P. Havinga, L. Van Hoesel, and J. Wu, “Collaborative algorithms for communication in wireless sensor networks,” in *Ambient Intelligence : Impact on Embedded Systems*, Kluwer Academic Publishers, ISBN 1-4020-7668-1, November 2003.
- [17] P. Havinga and L. Van Hoesel, “A lightweight medium access protocol LMAC for wireless sensor networks : Reducing preamble transmissions and transceiver state switches,” in *International Conference on Networked Sensing Systems (INSS)*, June 2004.
- [18] S. S. Chatterjea, L. Van Hoesel, and P. Havinga, “AI-LMAC : An adaptive, information-centric and lightweight MAC protocol for wireless sensor networks,” in *IEEE ISSNIP*, September 2004.
- [19] I. 802.11, “Carrier sense multiple access with collision detection CSMA/CD access method and physical layer specifications,” ANSI/IEEE, Standard 802.3, 1998.
- [20] —, “Part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” ANSI/IEEE, Standard 802.11 R2003, 1999.
- [21] P. Muhlethaler, *802.11 et les réseaux sans fil*. Eyrolles, 2002.
- [22] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *IEEE Infocom*, 2002, pp. 1567–1576.
- [23] —, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 783–791, June 2004.
- [24] T. Van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *ACM Sensys*, November 2003.
- [25] G. Lu, B. Krishnamachari, and C. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks,” in *Ad Hoc and Sensor Networks*, April 2004.
- [26] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *ACM Sensys*, November 2004.
- [27] G. P. Halkes, T. Van Dam, and K. G. Langendoen, “Comparing energy-saving MAC protocols for wireless sensor networks,” *Mobile Networks and Applications*, no. 10, pp. 783–791, 2005.

- [28] A. El-Hoiydil and J.-D. Decotigniel, "WiseMAC : An ultra low power MAC protocol for multi-hop wireless sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks*, ser. LNCS, vol. 3121. Springer Berlin / Heidelberg, 2004, pp. 18–31.
- [29] M. Awenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the efficiency of preamble sampling protocols for wireless sensor networks," in *Mobile Computing and Wireless Communications*, September 2006.
- [30] M. Buettnner, G. Yee, E. Anderson, and R. Han, "X-MAC : A short preamble mac protocol for duty-cycled wireless sensor networks," in *ACM Sensys*, November 2006.
- [31] N. Jongkeun, L. Sangsoon, and K. Chong-Kwon, "Dual wake-up low power listening for duty cycled wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2008, December 2008.
- [32] I. Rhee, A. Warriar, M. Aia, and M. J., "ZMAC : a hybrid MAC for wireless sensor networks," in *ACM Sensys*, November 2005.
- [33] I. Rhee, A. Warriar, J. Min, and L. Xu, "DRAND : Distributed randomized tdma scheduling for wireless adhoc networks," in *MobiHoc*, May 2006.
- [34] M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis, "Wireless sensor network energy-adaptive mac protocol," in *IEEE Consumer Communications and Networking Conference*, January 2006.
- [35] M. Brownfield, Y. Gupta, and N. Davis, "Wireless sensor network denial of sleep attack," in *IEEE Workshop on Information Assurance*, June 2005.
- [36] G. Ahn, E. Miluzzo, A. Campbell, S. Hong, and F. Cuomo, "Funneling-MAC : A localized, sink-oriented MAC for boosting fidelity in sensor networks," in *ACM Sensys*, November 2006.
- [37] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *ACM Sensys*, Los Angeles, California, USA, November 2003.
- [38] IEEE 802.15, "Part 15.4 : Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," ANSI/IEEE, Standard 802.15.4 R2006, 2006.
- [39] Zigbee, "Zigbee Specification," ZigBee Standards Organization, Standard Zigbee 053474r17, January 2008.
- [40] ZigBee Alliance, "The Zigbee Alliance web site," <http://www.zigbee.org/>.
- [41] A. Burr, *Modulation and Coding for Wireless Communications*. Pearson Education, 2001.
- [42] A. Koubaa, M. Alves, and E. Tovar, "A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks," in *WFCS*, July 2006, pp. 1–10.

- [43] —, “I-game : An implicit GTS allocation mechanism in IEEE 802.15.4 for timesensitive wireless sensor networks,” in *Euromicro Conference on Real-Time Systems*, 2006.
- [44] A. Van den Bossche, T. Val, and E. Campo, “Proposition and validation of an original MAC layer with simultaneous medium accesses for low latency wireless control/command applications,” in *IFAC WC*, 2008.
- [45] —, “Prototyping and performance analysis of a QoS MAC-layer for industrial wireless network,” in *IFAC FET*, 2007.
- [46] I. Chakeres and M. Belding-Royer, “On-demand TDMA scheduling for energy conservation in sensor networks,” in *International Conference on Distributed Computing Systems Workshops*, 2004.
- [47] IEEE 802.15 Task Group 4b, “TG4b contributions,” <http://grouper.ieee.org/groups/802/15/pub/TG4b.html>.
- [48] A. Koubaa, M. Alves, M. Attia, and A. Van Nieuwenhuyse, “Collision-free beacon scheduling mechanisms for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks,” Polytechnic Institute of Porto, Technical Report TR-061104, November 2006.
- [49] A. Koubaa, A. Cunha, and M. Alves, “A time division beacon scheduling mechanism for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks,” Poly. Institute of Porto, Tech. Rep. TR-070401, April 2007.
- [50] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad hoc networks*, vol. 3, pp. 325–349, 2005.
- [51] J. Ibriq and I. Mahgoub, “Cluster-based routing in wireless sensor networks : issues and challenges,” in *SPECTS*, 2004.
- [52] S. Mahfoudh and P. Minet, “EOLSR : an energy efficient routing protocol in wireless ad hoc and sensor networks,” *Journal of Interconnection Networks*, vol. 9, no. 4, 2008.
- [53] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized link state routing protocol,” IETF, RFC 3626, 2003.
- [54] S. Mahfoudh and P. Minet, “Maximization of energy efficiency in wireless ad hoc and sensor networks with SERENA,” *Journal on Mobile Information System*, to appear.
- [55] A. Guitton, F. Delobel, G. Chalhoub, and M. Misson, “Optimal algorithm for the minimal ordering of a tree-based diffusion,” Clermont university, Internal Report, February 2009.
- [56] G. Chalhoub, A. Guitton, F. Delobel, and M. Misson, “Implementation details for the lower layers of an LP-WPAN simulation framework,” *International Journal of Wireless Communication and Simulation (IJWCS)*, 2009.

- [57] H. Friis, "A note on a simple transmission formula," in *Proceedings of the IRE*, May 1946.
- [58] T. S. Rappaport, *Wireless communications : principles and practice*. Prentice Hall, 2002.
- [59] ITU, "Propagation data and prediction method for the planning of indoor radio communication systems and local area networks in the frequency range of 900 MHz to 100 GHz," ITU, Recommendation ITU-R P 1238-4, 2005.
- [60] S. Benferhat, M. Misson, and F. Jacquet, "Study of traffic conditions for a fleet of mobiles moving within a wireless cell in an industrial environment," in *MIC-CCA, Mosharaka International Conference on Communications, Computers and Applications*, 2008.
- [61] Chipcon, "Chipcon CC2420," <http://focus.ti.com/docs/prod/folders/print/cc2420.html>.
- [62] Motorola, "Motorola MC9S08GT60A," [http://www.digchip.com/datasheets/new\\_parts/200808/-1127519.php](http://www.digchip.com/datasheets/new_parts/200808/-1127519.php).
- [63] Freescale, "Codewarrior," <http://www.freescale.com/codewarrior>.
- [64] G. Chalhoub, A. Guitton, F. Jacquet, A. Freitas, and M. Misson, "Medium access control for a tree-based wireless sensor network : Synchronization management," in *IFIP Wireless Days*, 2008.
- [65] T. R. Park, T. H. Kim, J. Y. Choi, S. Choi, and W. H. Kwon, "Throughput and energy consumption analysis of IEEE 802.15.4 slotted CSMA/CA," *Electronics Letters*, vol. 41, no. 18, pp. 1017–1019, September 2005.
- [66] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. Van Der Perre, F. Catthoor, I. Moerman, A. Bahai, and P. Varaiya, "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer," in *IEEE Globecom*, November 2006, pp. 1–6.
- [67] Z. Tao, S. Panwar, D. Gu, and J. Zhang, "Performance analysis and a proposed improvement for the IEEE 802.15.4 contention access period," in *WCNC*, April 2006, pp. 1–8.
- [68] Z. Jie and L. HongLi, "Access delay analysis of ZigBee protocol with delay line," in *WICOM*, September 2007, pp. 6452–6455.
- [69] F. Shu and T. Sakurai, "Analysis of an energy conserving CSMA-CA," in *IEEE Globecom*, November 2007, pp. 2536–2540.
- [70] J. Zheng and M. J. Lee, *A Comprehensive Performance Study of IEEE 802.15.4*. IEEE Press, 2004.
- [71] IEEE 802.15, "Part 15.4 : Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)," ANSI/IEEE, Standard 802.15.4 R2003, 2003.

- [72] G. Chalhoub, N. Haddid, A. Guitton, and M. Misson, “Deference mechanisms significantly increase the MAC delay of slotted CSMA/CA,” in *ICC, International Conference on Communications*, June 2009.
- [73] O. Sharon and E. Altman, “An efficient polling MAC for wireless LANs,” *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 4, August 2001.
- [74] G. Chalhoub, E. Livolant, A. Guitton, A. van den Bossche, M. Misson, and T. Val, “Specifications and evaluation of a MAC protocol for a LP-WPAN,” *Ad Hoc & Sensor Wireless Networks journal*, 2009.

Un réseau de capteurs sans fil est un ensemble de noeuds communicants, généralement miniatures et capables de fonctionner avec peu d'énergie. Certaines applications imposent à ces réseaux de capteurs sans fil d'être économes en énergie pour prolonger la durée de vie du réseau, d'autres nécessitent un comportement déterministe qui se traduit généralement par des délais de traversée du réseau bornés et/ou un taux de trames perdues inférieur à un seuil critique. Pour économiser significativement de l'énergie, une entité doit se mettre en mode sommeil. Durant cette période, elle sera inactive donc incapable de participer à l'activité du réseau. L'usage de périodes de sommeil est un handicap pour l'aspect déterministe et pour le respect de contraintes temporelles.

Le travail de cette thèse est centré sur les spécifications, le développement et l'évaluation sous différentes formes d'une méthode d'accès au médium adaptée au déterminisme et aux exigences de qualité de service telles que l'absence de collisions et la garantie d'un délai borné de bout-en-bout, tout en conservant une faible consommation énergétique. La solution proposée repose sur la segmentation temporelle des activités, une synchronisation multi-sauts et une différenciation de services basée sur des stratégies de routage adaptées.

**Mots clés :** Réseaux de capteurs sans fil, Méthodes d'accès, ZigBee, IEEE 802.15.4, Qualité de service.

A wireless sensor network is a set of small communicating nodes that consume very little energy. Some applications require wireless sensor networks to be energy efficient to extend the lifetime of the network, others require deterministic behavior that usually results in bounded end-to-end delays and/or frame loss rate below a critical threshold. To save significant energy, an entity must be put into sleep mode. During this period, it will be inactive thus unable to participate in network activity. The use of sleep periods is a handicap for the deterministic aspect as well as for the respect of delay constraints.

The work of this thesis focuses on the specification, development and evaluation in different forms of a deterministic medium access mechanism adaptable to the quality of service requirements such as the absence of collisions and the guarantee of a bounded end-to-end delay, while maintaining low energy consumption behavior. The proposed solution is based on temporal segmentation of the activities of the nodes, multi-hop synchronization and service differentiation based on two routing strategies.

**Keywords:** Wireless sensor networks, Medium access control, ZigBee, IEEE 802.15.4, Quality of service.